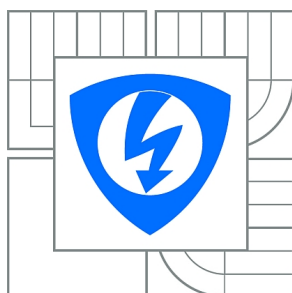




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ELEKTRONICKÝ SYSTÉM ZAVLAŽOVÁNÍ

ELECTRONICAL IRRIGATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VÁCLAV ENDRYCH

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ZDENĚK BRADÁČ, Ph.D.

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Václav Endrych

ID: 106421

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Elektronický systém zavlažování

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncepci elektronického zavlažovacího systému. Systém navrhněte jako 4 okruhový systém pro řízení zavlažování ve čtyřech nezávislých větvích. Navrhněte elektroniku na základě mikrokontroléru Rabbit s rozhraním Ethernet/WiFi. Doplňte o vstupní snímač vlhkosti. Navrhněte elektroniku jednotky, navrhněte a realizujte DPS, osadte ji a oživte systém. Vybavte programovým vybavením pro mikrokontrolér Rabbit i WWW stránkami pro nastavování a konfiguraci. Demonstrujte správnou funkci zavlažovacího systému.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6
Dle pokynů vedoucího práce.

Termín zadání: 7.2.2011

Termín odevzdání: 30.5.2011

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato bakalářská práce pojednává o automatických zavlažovacích systémech, jejich účelu, běžném provedení a o komponentách, z nichž jsou sestaveny, zejména pak o elektronických řídicích jednotkách. Cílem této práce bylo zkonstruovat právě takovou řídicí jednotku. V práci je popsán postup návrhu i realizace tohoto zařízení od stanovení výchozích požadavků, přes návrh obvodového zapojení a fyzické konstrukce až po návrh struktury řídicího programu a zprovoznění zařízení. Výsledkem práce je funkční řídicí jednotka pro čtyřokruhový zavlažovací systém s možností měření vlhkosti ovzduší čtyřmi nezávislými čidly. Jednotka poskytuje množství různých funkcí a je ovládána na dálku přes Ethernetovou přípojku.

Klíčová slova

zavlažovací systém, řídicí jednotka, procesor Rabbit 3000, webové rozhraní, SSI, CGI

Abstract

This bachelor thesis is concerned with automatic irrigation systems. It describes their purpose their usual fabrication and their individual components, especially electronic controlling units. The goal of this thesis was to construct such unit. Thesis furthermore describes the process of design and realization of this device, starting from determination of the basic requirements on the device, through design of its circuitry and construction, up to design of included control program and final tuning of the device. Resulting controlling unit is functional. It can drive four branches of an irrigation system and measure relative humidity by four independent sensors. Unit offers a number of different functions and is controlled from remote user interface via an Ethernet connection.

Keywords

irrigation system, controlling unit, processor Rabbit 3000, web interface, SSI, CGI

Bibliografická citace:

ENDRYCH, V. *Elektronický systém zavlažování*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 2011. 73 s. Vedoucí bakalářské práce: doc. Ing. Zdeněk Bradáč, Ph. D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Elektronický systém zavlažování jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne:

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Zdeněku Bradáčovi, Ph. D. za odborné a pedagogické vedení, za konstruktivní kritiku a další rady při zpracování mé bakalářské práce.

V Brně dne:

.....

podpis autora

OBSAH

ÚVOD	8
1 AUTOMATICKÉ ZAVLAŽOVACÍ SYSTÉMY	9
1.1 PODOBA A ÚČEL	9
1.1.1 <i>Výhody a možnosti</i>	9
1.1.2 <i>Instalace a obsluha</i>	10
1.2 JEDNOTLIVÉ KOMPONENTY	11
1.2.1 <i>Rozvody</i>	11
1.2.2 <i>Postřikovače</i>	11
1.2.3 <i>Elektroventily</i>	12
1.2.4 <i>Snímače</i>	13
1.2.5 <i>Řídicí jednotky</i>	14
1.3 OBVYKLÉ FUNKCE A MOŽNOSTI ŘÍDICÍCH JEDNOTEK	15
2 NÁVRH FYZICKÉ REALIZACE JEDNOTKY	16
2.1 VÝCHOZÍ POŽADAVKY	17
2.2 PARAMETRY FUNKČNÍCH ČÁSTÍ JEDNOTKY	18
2.2.1 <i>Řídicí modul</i>	18
2.2.2 <i>Vstupy jednotky</i>	19
2.2.3 <i>Výstupy jednotky</i>	19
2.2.4 <i>Přímé ovládání</i>	19
2.3 OBVODOVÉ ŘEŠENÍ	21
2.3.1 <i>Řídicí modul</i>	21
2.3.2 <i>Vstupy jednotky</i>	22
2.3.3 <i>Výstupy jednotky</i>	23
2.3.4 <i>Přímé ovládání</i>	23
2.3.5 <i>Napájecí zdroj</i>	25
2.3.6 <i>Shrnutí</i>	26
2.4 KONSTRUKČNÍ ŘEŠENÍ	28
2.4.1 <i>Třívrstvá struktura</i>	28
2.4.2 <i>Účelové rozmístění součástek</i>	29
2.4.3 <i>Finální forma</i>	29
2.4.4 <i>Shrnutí</i>	30

OBSAH *(Pokračování)*

3 NÁVRH ŘÍDICÍHO PROGRAMU JEDNOTKY	32
3.1 NÁVRH ZÁKLADNÍ STRUKTURY PROGRAMU	32
3.1.1 Uspořádání hlavních procesů	32
3.1.2 Výsledné časování	34
3.1.3 Koncepce ovládacích rozhraní	34
3.1.4 Paralel Port F Bug	35
3.2 HLAVNÍ PROCESY	36
3.2.1 Buttons Handler	36
3.2.2 HTTP Handler	37
3.2.3 Task Handler	38
3.3 WEBOVÉ OVLÁDACÍ ROZHRANÍ	43
3.3.1 HTML Stránky	43
3.3.2 CGI Funkce	44
3.4 OVLÁDACÍ MENU JEDNOTKY	48
3.4.1 Menu „Handler“	48
3.4.2 Koncepce ovládacího menu	51
3.5 SHRNUÍ	54
ZÁVĚR	56
LITARATURA A ZDROJE INFORMACÍ	57
SEZNAM POUŽITÝCH ZKRATEK	58
SEZNAM OBRÁZKŮ	59
SEZNAM PŘÍLOH	60

ÚVOD

Cílem této bakalářské práce bylo navrhnout, zrealizovat a zprovoznit elektronickou řídicí jednotku pro zavlažovací systém. Jednotka měla být schopna ovládat 4 nezávislé okruhy zavlažování a být ovládána přes webové rozhraní. Práce je rozdělena na tři části.

První část práce poskytuje stručný přehled o již existujících komerčních řešeních systémů automatického elektronického zavlažování, o jejich účelu jako takovém a o jejich nejběžnějších provedeních. Poskytuje také přehled o jednotlivých komponentách zavlažovacích systémů a jejich parametrech.

Druhá a třetí část práce pojednává o postup návrhu a konečném řešení zkonstruované řídicí jednotky. Část druhá se zabývá návrhem obvodového a konstrukčního řešení technické části jednotky od počátečního stanovení požadavků až po fyzickou konstrukci. Část třetí se zabývá návrhem a provedením programového vybavení jednotky.

1 AUTOMATICKÉ ZAVLAŽOVACÍ SYSTÉMY

Tato kapitola obsahuje základní pohled na to, co je to automatický zavlažovací systém, jaký je jeho účel a jaké jsou důvody pro jeho použití. Dále poskytuje stručný přehled o průběhu jeho instalace, o běžném chodu automatického zavlažovacího systému a základní informace o komponentách, z nichž jsou sestavovány nejběžnější zavlažovací systémy, jejich možnosti parametry*.

1.1 Podoba a účel

Automatický zavlažovací systém je klasicky potrubní systém rozvodů a postřikovačů umístěný pod povrchem trávníku (zahrady). K tomuto systému dále patří i všechny komponenty zajišťující jeho automatickou činnost bez nutnosti zásahu člověka. Zejména pak ovládací elektroventily, vstupní čerpadla, elektrické rozvody a elektronická řídicí jednotka, včetně napájecího modulu. Systém je také možno doplnit o různá čidla podnebních podmínek, aby tak řídicí jednotka mohla přizpůsobit program zavlažování aktuálnímu počasí.

„Automatický zavlažovací systém se v současné době stává již samozřejmou součástí většiny okrasných zahrad a trávníků. Zahradu, jako místo pro zábavu a odpočinek, si bez stále zeleného a svěžího trávníku nelze představit. Stejně tak jako trávník bez vody.“ [1]

Automatický zavlažovací systém je téměř nutností u velkých zelených prostorů určených veřejnosti nebo ke komerčním účelům, u menších soukromých prostor, jako je například zahrada rodinného domu, je pak velkou předností.

1.1.1 Výhody a možnosti

Jednoznačnou výhodou systému je šetření času a práce se zavlažováním (popřípadě tedy i snížení nákladů firmy spojených s udržováním trávníku). Dále odpadá nutnost roztahování a uklízení přírodních hadic při každé závlaze, kromě dalšího ušetření práce je tak samotné zavlažování i značně estetičtější. Další, ne tak zjevnou, ale o to důležitější, výhodou je vysoce kvalitní závlaha. Při dobrém vypracování projektu systém zavlažuje velmi rovnoměrně celou plochu povrchu. Pomocí řídicí jednotky je možné velmi přesně seřizovat délku a periodu závlahy, tyto parametry se navíc můžou automaticky měnit podle aktuální sezóny a ušetřit tak i na spotřebě vody.

* Informace a obrázky obsažené v této kapitole jsou většinou převzaty ze serveru AZzahrada.cz

V neposlední řadě je možné zavlažovací systém vyhotovit tak, že všechny důležité komponenty jsou umístěny v uzamykatelných skříňkách a postřikovače v neaktivním stavu zapadnou a splynou s povrchem trávníku (a prakticky téměř nejsou k nalezení). Takový systém je pak vhodný zejména pro otevřené prostory přístupné veřejnosti.

U rozlehlých prostor, je možné se vyvarovat složitým a nepřehledným elektrickým rozvodům použitím komponent s moduly radiové komunikace. Napájení jednotlivých komponent je pak realizováno pomocí kvalitních akumulátorů. Systém lze realizovat i decentralizovaně užitím malých řídicích modulů umístěných přímo u ovládacích elektroventilů.

1.1.2 Instalace a obsluha

Vyhotovení zavlažovacího systému a všechny práce u finálního zákazníka zpravidla provádějí pracovníci firmy autorizované výrobcem komponent pro závlahové systémy. Tato firma potom ručí za správnou instalaci všech komponent, kvalitní provedení celého zavlažovacího systému a udržení z toho vyplívající vysoké životnosti systému. Firma navíc poskytuje všechny související služby jako je vypracování projektu a projektové dokumentace, zajištění stavebního povolení a dodání návodu k obsluze. Dále je pak k dispozici například záruční a havarijní servis, nebo vyhotovení videozáznamu realizace systému. Naopak ale existují i firmy, které nabízejí možnost realizace systému svépomocí zákazníka, čímž jsou značně sníženy finanční pořizovací náklady systému.

Běžná obsluha systému je pak v ideálním stavu zúžena pouze na případné změny nastavení zavlažování na ovládání řídicí jednotky. Při nutnosti větších úprav nebo oprav je (při použití doporučených šachtic) vždy rychlý a jednoduchý přístup k ovládacím elektroventilům a ručním uzávěrům a postřikovače lze snadno rozebrat pomocí přiložených montážních klíčů (například kvůli výměně trysky). Velké úpravy a zejména výkopové práce jsou potřeba jedině až při poruše nebo nutnosti přesunutí trubního vedení.

1.2 Jednotlivé komponenty

Tato kapitola obsahuje základní přehled jednotlivých komponent, z nichž jsou automatické zavlažovací systémy běžně sestavovány. Pro každou komponentu jsou uvedeny základní informace a parametry.

Na konci kapitoly je uveden přehled běžných možností a funkcí řídicích jednotek zavlažovacích systémů.

1.2.1 Rozvody

Pro rozvod vody na zavlažování se používají polyetylenové hadice s průměrem 20 až 63 mm. Pro spojování a rozbočování zpravidla slouží plastové spojky, kolena a T-kusy. Jejich montáž je velmi rychlá a postačuje k ní utahovací klíč. Rozebírání je také velmi snadné, spoj je přesto pevný a spolehlivý.

1.2.2 Postřikovače

Velmi důležitými komponentami zavlažovacího systému jsou samotné postřikovače. Mnoho parametrů zavlažování je totiž určeno právě výběrem a mechanickým nastavením postřikovačů. Klasické rotační postřikovače jsou zpravidla výsuvné a mají vodomazný pohon, jejich nastavení se pak často provádí jednoduše instalací jedné z výměnných trysek (ze sady dodávané výrobcem současně s postřikovačem) pomocí dodaného montážního klíče. Takto lze pak nastavit úhel zavlažovací výseče (popřípadě celý kruh), délku dostřiku a úhel vzestupu zavlažovacího paprsku.

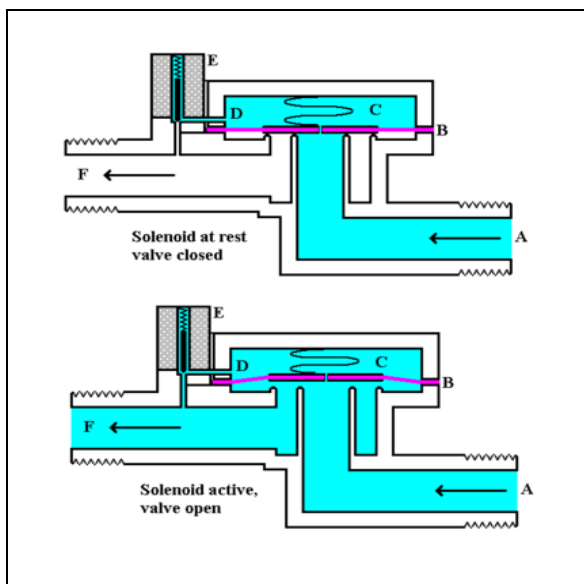


Obrázek 1.1 - Ukázka rotačního postřikovače - Hunter PGM

Lepší postřikovače disponují i manuální regulací průtoku či možností úplného zastavení (je tak možno vyměnit trysku postřikovače i bez vypínání celého zavlažovacího systému). Postřikovače určené pro větší plochy jsou navíc doplněny o jednu nebo dvě rozprašovací trysky pro rovnoměrné zavlažování až do vzdálenosti 20 metrů i více. Dalšími důležitými parametry postřikovačů jsou například pracovní tlak, rychlost rotace, průměr a výška výsuvníku, odolnost vůči znečištění a spotřeba vody vzhledem k zavlažované ploše.

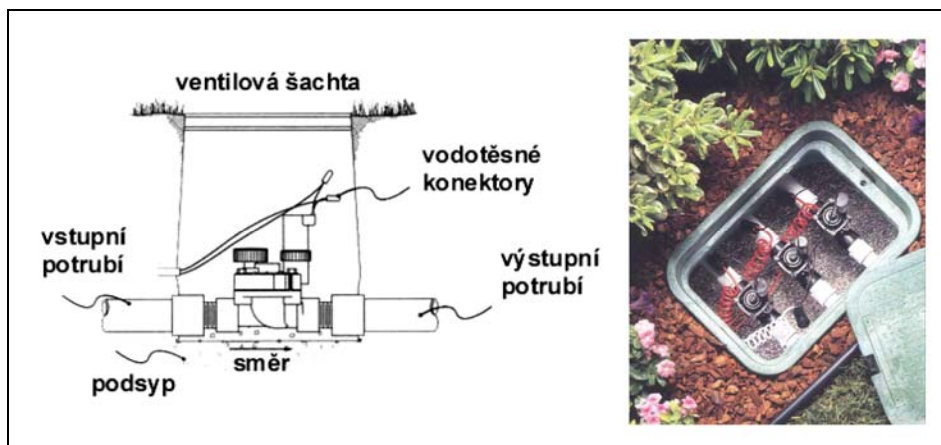
1.2.3 Elektroventily

Pro elektronické řízení zavlažovacího systému je využíváno elektroventilů ovládaných solenoidní cívkou. Princip fungování ventilu je znázorněn na obrázku **Obrázek 1.2**. Lepší ventily mají možnost regulace průtoku a manuálního ovládání. U některých ventilů je i výměnná ovládací cívka.



Obrázek 1.2 - Funkce elektroventilu

Při správné instalaci vodotěsných konektorů ovládací cívky může ventil pracovat prakticky i pod vodou. Stále se ale silně doporučuje umístění ventilů ve speciální (uzamykatelné) šachtici. K ventilům je tak dobrý přístup při kontrolách a případných servisních opravách, současně jsou ale ventily chráněny proti nepovolaným osobám a povětrnostním vlivům. Zvyšuje se tak životnost ventilů. Šachtice se zesílenou konstrukcí lze umístit i na frekventované veřejné plochy.



Obrázek 1.3 - Správná instalace elektroventilů

1.2.4 Snímače

Pro doplnění zavlažovacího systému jsou k dostání snímače informující řídicí jednotku o aktuálních podnebných podmínkách. Je možné použít buď samostatné snímače větru, srážek a teploty nebo malou meteostanici zahrnující všechny snímače. Součástí snímačů je dlouhý přívodní kabel pro připojení na řídicí jednotku, snímače pak většinou fungují pouze v blokovacím režimu (například pokud prší, je zablokován zavlažovací program apod.). Některá čidla umožňují seřízení vypínací/zapínací úrovně.

Čidla je potřeba umístit do volného prostoru nad úroveň zemského povrchu. Dále je doporučeno umístění na slunném místě, kde dojde k rychlému oschnutí po dešti. Součástí snímačů je zpravidla i upevňovací konzole.

1.2.5 Řídicí jednotky

Řídicí jednotky jsou většinou koncipovány tak, aby jejich ovládání bylo jednoduché a přehledné, současně však často poskytují velké množství funkcí. V klasickém uspořádání má jednotka velký centrální přepínač programů, popřípadě i malý přepínač funkce vstupních čidel, jednoduchý LCD display s několika číslicemi a grafickými ikonkami a několik tlačítek pro ovládání jednotlivých nastavení.

U jednotek se předpokládá instalace přímo na zeď nebo do uzamykatelného boxu. K napájení malých jednotek je nečastěji použit externí napájecí modul 24V DC, větší jednotky mají tento modul již vestavěný ve skříni jednotky. Jednotky obsahují i záložní zdroj a nízkonapěťové paměti pro uchování dat programů při výpadku proudu.

Malé jednotky jsou schopny ovládat 2 až 6 sekcí (jeden hlavní elektroventil nebo čerpadlo současně s jedním sekčním ventilem). Větší jednotky pak již mají modulární řešení výstupu, například: základem je ovládání osmi sekcí s možností dodat až 3 výstupní moduly pro 8 sekcí, celkem tedy $4 \times 8 = 32$ sekcí (ovládání jednoho hlavního elektroventilu nebo čerpadla současně s jedním nebo dvěma sekčním ventily). I větší jednotky tedy spíše počítají se zavlažováním vždy jen v jedné sekci v daném okamžiku. Lepší jednotky mají i 9V akumulátor a lze je pak úplně vyjmout ze skříně a programovat přímo na pozemku.



Obrázek 1.4 - Ukázka řídicí jednotky - Hunter EC

1.3 Obvyklé funkce a možnosti řídicích jednotek

- nastavení 12h nebo 24h formátu času
- indikace probíhající závlahy
- nastavení délky závlahy od 1 min do několika hodin (po 1min krocích)
- 3 nezávislé programy zavlažování
- každý program umožňuje až 4 denní starty (celkem 12 startů)
- sedmidenní kalendář s volbou zavlažovacích dní
- volba zavlažování v sudé nebo liché dny
- volba zavlažovací pauzy mezi spouštěním jednotlivých sekcí
- manuální spouštění jedné sekce nebo skupiny sekcí
- manuální blokování programu
- manuální zadání přestávky v závlaze 1 až 31 dní
- manuální změna délky závlahy bez změny uložených dat 10 ÷ 150 %

2 NÁVRH FYZICKÉ REALIZACE JEDNOTKY

Tato kapitola obsahuje shrnutí návrhu technické části řídicí jednotky pro automatický zavlažovací systém, která je předmětem této práce. Podkapitoly jsou řazeny chronologicky podle postupných fází návrhu.

Na úvod jsou shrnuty výchozí požadavky kladené na řídicí jednotku. V další kapitole jsou pak uvedeny základní parametry jednotlivých funkčních bloků výsledného zařízení, které vyplynuly právě z úvodních požadavků na jednotku. Následuje popis navrženého obvodového řešení řídicí jednotky a popis návrhu konstrukčního uspořádání zařízení.

Podkapitoly 3 a 4 jsou zakončeny shrnutím výsledných parametrů a možností jednotky určených danou fází návrhu zařízení a také shrnutím technických nedostatků zjištěných po skutečné konstrukci zařízení.

2.1 Výchozí požadavky

Od počátku práce byly na řídicí jednotku stanoveny tyto požadavky:

- Jádrem jednotky bude řídicí modul Rabbit RCM3200
- Jednotka bude schopná ovládat 4 nezávislé okruhy zavlažování
- Řízení bude vycházet zejména z reálného času
- Jako doplňkový vstup bude měřena vlhkost ovzduší
- Výstupem bude spínání solenoidních ventilů
- Návrh jednotky bude sestávat z jednoduše dostupných součástek

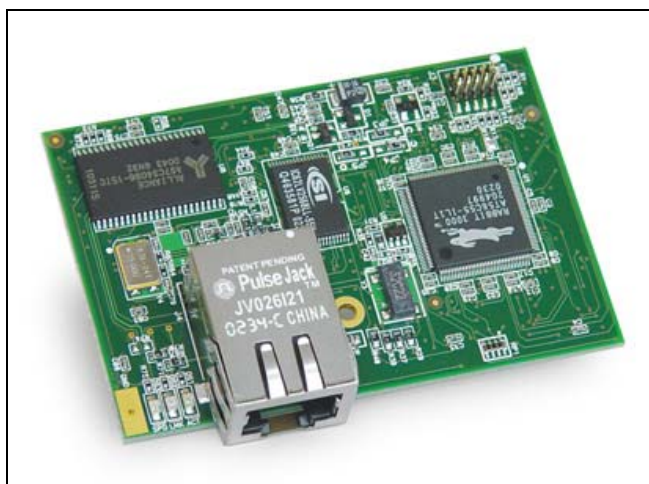
V průběhu práce pak byly doplněny ještě tyto rozšiřující požadavky:

- Jednotka bude umožňovat i přímé ovládání pomocí 4 směrových tlačítek a alfanumerického LCD displeje 4x20 znaků (modul CM200400SFAYA-02).
- Na výstup bude možno připojit libovolný ventil do jmenovitého napětí 250V AC.
- Jednotka bude umožňovat připojení čtyř nezávislých čidel vlhkosti (jedno pro každý okruh zavlažování)

2.2 Parametry funkčních částí jednotky

2.2.1 Řídicí modul

Řídicím jádrem celé jednotky je tedy modul RCM3200 s procesorem Rabbit 3000. Modul již obsahuje krystalový oscilátor (budící procesor na frekvenci 44,2 MHz), paměti 512 kB Flash, 256 kB Data SRAM, 512 kB Program Execution SRAM, reset generátor, obvody pro Ethernetovou komunikaci a přípojku RJ-45. Uživatel modulu musí poskytnout pouze napájení +3,3V. Dále je vhodné připojit vstup resetu (typicky tlačítko) a záložní baterii (udržuje v chodu obvod reálného času a obsah Datové SRAM paměti). Modul se připojuje přes dva konektory 2x17 pinů s rozestupem 2 mm na spodní straně desky, kde jsou vyvedeny jak základní funkční signály (napájení, reset, apod.), tak jednotlivé vstupní a výstupní piny sdružené do portů.



Obrázek 2.1 - Řídicí modul Rabbit RCM3200

Procesor na modulu je možné programovat přes konektor 2x5 pinů, s rozestupem 1,27 mm, na horní straně modulu za použití příslušného programovacího kabelu. Kabel také umožňuje krokování programu v procesoru a režim diagnostiky. Všechny režimy je možné použít za chodu zařízení, tedy s modulem připojeným k uživatelské desce. [7]

2.2.2 Vstupy jednotky

Vstupem jednotky (řídícího procesoru) mají být hodnoty vlhkosti ovzduší změřené připojenými čidly vlhkosti (jedno čidlo pro každý okruh). Pro měření vlhkosti existují nejrůznější již hotové měřicí moduly, jednoduchá čidla s proměnnou rezistivitou nebo kapacitou (musí být napájena střídavým proudem) až po miniaturní inteligentní čidla kalibrovaná z výroby se sériovým výstupem.

Pro účely této práce se předpokládá použití čidel s lineárním napěťovým výstupem, jako je například integrované čidlo H1H-4000 od firmy Honeywell. Pro každé čidlo je vyvedeno napájecí napětí 5V (a uzemnění).

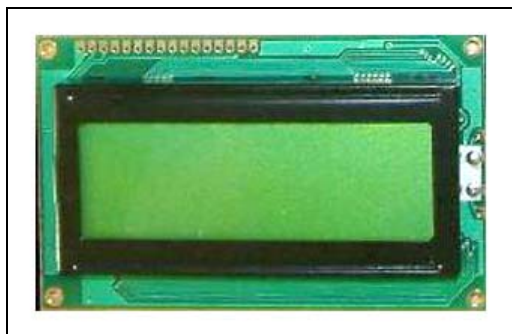
2.2.3 Výstupy jednotky

Účelem jednotky je elektronické ovládání čtyř okruhů zavlažování. V tomto případě se předpokládá použití čtyř solenoidních ventilů (elektroventilů ovládaných solenoidní cívkou). Tyto elektroventily jsou zpravidla k dostání s cívkami s napájením od 9 do 24 voltů, na střídavý proud. Existují však například i bistabilní ventily se 6V pulsním ovládním.

Dle rady vedoucího práce jsou skutečným výstupem jednotky čtyři elektromagnetická spínací relé. Tato varianta výstupu umožňuje připojení jakéhokoliv elektroventilu s napájením do 250V/10A AC. Dále umožňuje i připojení různých ventilů pro každý okruh. Uživatel řídící jednotky však musí pro každý ventil sám poskytnout a zapojit příslušný napájecí zdroj. Výstup také umožňuje pouze jednoduché On/Off ovládání ventilů.

2.2.4 Přímé ovládání

Přestože jednotku bude možno plně ovládat přes Ethernetové připojení, v průběhu návrhu byl na jednotku doplněn požadavek i na přímé ovládání pomocí čtyř směrových tlačítek a alfanumerického LCD displeje. Pro zobrazování byl vybrán LCD modul s řadičem CM200400SFAYA-02.



Obrázek 2.2- Zobrazovací modul CM200400SFAYA-02

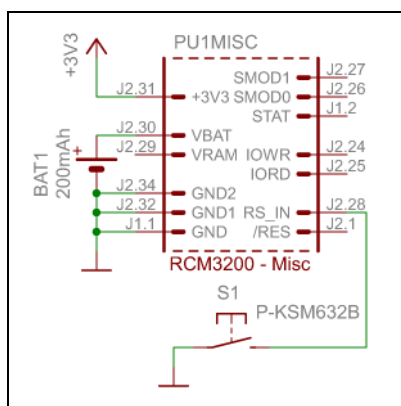
Modul má 4x20 znaků, každý z nich 5x8 bodů. Jmenovité napájecí napětí řadiče displeje je 5 V. Napájení samotného LCD displeje je o něco menší, přibližně 4,5 V. Displej má podsvícení (LED pole) napájené z odděleného okruhu. Řízení modulu probíhá přes osmibitovou paralelní sběrnici a tři řídicí signály RS, R/W a E. Bit R/W určuje operaci čtení nebo zápisu, bit RS určuje zápis instrukce nebo dat a bit E (Enable) spouští danou operaci. [6]

2.3 Obvodové řešení

2.3.1 Řídicí modul

K řídicímu modulu je jako funkční základ připojena zem a napájení 3,3 V, modul typicky spotřebovává proud 145 mA (při nezatížených výstupech) plus dalších 10 mA při připojení programovacího kabelu. Spotřebu modulu je možné omezit snížením taktovací frekvence procesoru. Pokud je v provozu i Ethernetový řadič, modul může celkově spotřebovávat až 255 mA [7].

Dále je připojena záložní baterie a tlačítko Reset. Signál RESET_IN je na řídicím modulu připojen přes odpor 10 k Ω na napájení, vně modulu tedy stačí zapojit pouze dolní větev - tlačítko na zem. Při nezapojeném hlavním napájení modul ze záložní baterie typicky spotřebovává proud 12 μ A. Výrobce doporučuje použití lithiové baterie (která udržuje téměř konstantní nominální napětí 3 V po většinu doby své životnosti) s minimální kapacitou 165 mAh. [7] Zde je použita lithiová baterie CR2032 s kapacitou 200 mAh.



Obrázek 2.3 - Základní připojení řídicího modulu

Co se týče vstupů a výstupů řídicího modulu (procesoru), nebyly v tomto návrhu využity pouze porty D (port je na modulu použit ke komunikaci přes Ethernetovou přípojku) a port C (port sériových linek, který má střídavě pouze vstupní a pouze výstupní piny).

Port A (paralelní datová sběrnice) je kompletně využit pro komunikaci se zobrazovacím LCD modulem společně s portem F, na který jsou připojeny řídicí signály zobrazovacího modulu a řídicí signály komunikace s ním. Port B ovládá výstupní relé, Port G vstupní A/D převodníky a k portu E jsou připojena ovládací tlačítka. Zapojení budou podrobněji rozvedena v následujících kapitolách.

2.3.2 Vstupy jednotky

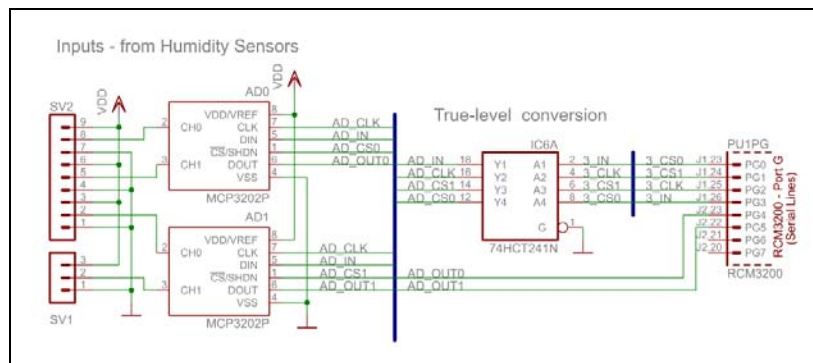
Pro měření vlhkosti je počítáno s lineárním napěťový vstupem ($U_{in} = k \times RH$) a je možno připojit až 4 čidla vlhkosti (jedno čidlo pro každý okruh). Použitý řídicí procesor má ale pouze digitální vstupy, bylo proto potřeba na vstup zařadit A/D převodníky.

Původně měl být použit čtyřkanálový osmibitový převodník ADC0834, který umožňuje přímé ovládání pomocí signálů s True-level 3,3V. Nicméně v průběhu návrhu jednotky byl tento převodník vyřazen z nabídky dodavatele součástek, proto jsou nakonec použity dva dvoukanálové převodníky MCP3202-BI/P, které jsou lépe k dostání.

Převodníky disponují integrovaným Sample&Hold odvodem, dvanáctibitovou přesností a dobou převodu 10 μ s. Výstupem převodníků je sériové SPI rozhraní. Převodník má sice korektně pracovat již od napájecího napětí 2,7 V, ale při předpokladu 5V vstupního signálu je potřeba napájet převodník z 5V zdroje*. V důsledku toho má převodník na řídicích signálech příliš vysoký vstupní práh a pro správnou komunikaci mezi procesorem a převodníky bylo třeba zapojení doplnit o převod napěťových úrovní.

Tento převod je realizován pomocí integrovaného obvodu 74HCT244, který obsahuje dva čtyřbitové jednosměrné neinvertující budiče. Tento obvod má snížený vstupní práh, který umožňuje připojení vstupního signálu s True-level 3,3 V, výstupní signál pak již správně odpovídá vstupním prahům A/D převodníků. Druhý budič z obvodu byl navíc posléze použit pro komunikaci s LCD zobrazovacím modulem. Výstupy z A/D převodníků už není třeba převádět, protože na vstupy řídicího procesoru lze připojit pětivoltový signál (jsou 5V Tolerant).

* Převodník umožňuje použití referenčního napětí pro analogové vstupy, toto napětí ale smí být pouze menší než jeho napájecí napětí a lze tedy použít pouze ke zvýšení vstupní citlivosti převodníku. Proto musí být převodník napájen pěti volty a nelze použít napájení 3,3V a referenční napětí 5V.

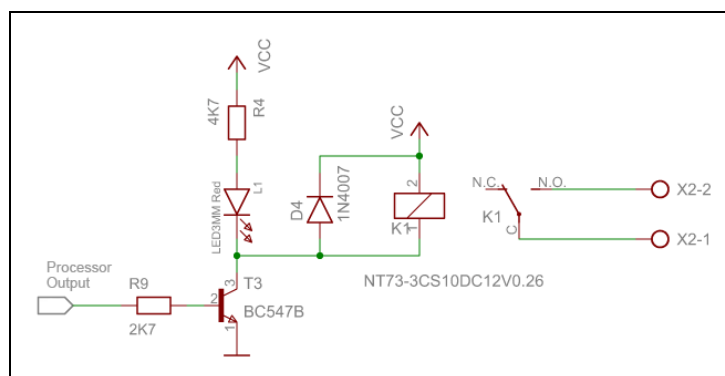


Obrázek 2.4 - Vstupní obvod

2.3.3 Výstupy jednotky

Na výstupu jednotky jsou čtyři elektromagnetická relé, z nichž každé umožňuje spínat zátěž do 250V/10A AC. Cívky relétek spínají při 12 V a mají odpor 400 Ω . Při sepnutí tedy spotřebovávají proud 30 mA (120 mA celkem při sepnutí všech čtyř výstupů).

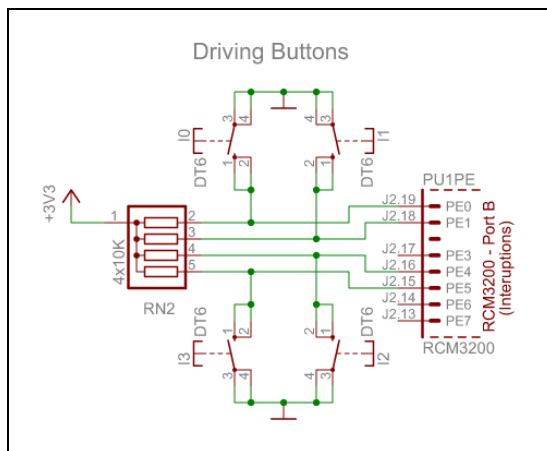
Procesor ovládá výstupní relátka (proudy cívkami) přes signálové tranzistory BC547B ve spínacím režimu. Tranzistorům je pro snížení saturačního napětí vnuceno proudové zesílení přibližně 25x. Zapojení každého relé je doplněno ještě o zpětnou diodu a kontrolku sepnutí.



Obrázek 2.5 - Výstupní obvod (jeden výstup)

2.3.4 Přímé ovládání

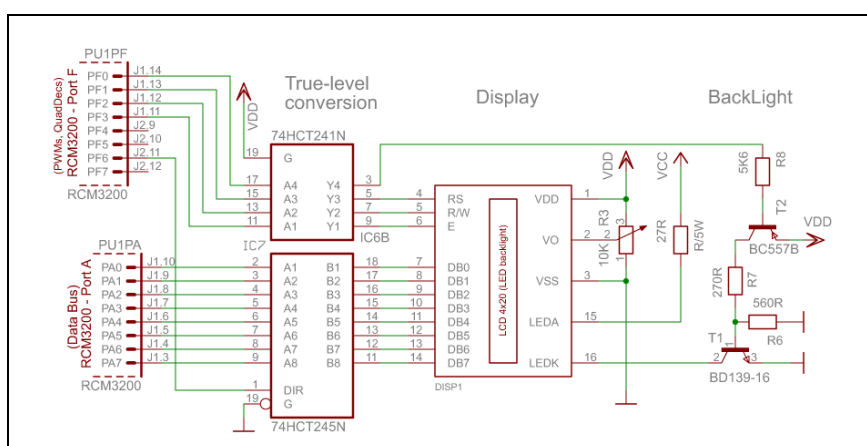
Pro přímé ovládání je využito čtyř tlačítek připojených přímo na vstupy procesoru a zobrazovacího LCD modulu CM200400SFAYA-02 pro zobrazení dat a komunikaci s uživatelem.



Obrázek 2.6 - Ovládací tlačítka

Napájení řadiče displeje je připojeno přímo na 5V zdroj, logika řadiče spotřebovává proud 2 mA. Napájení samotného displeje je přepojeno přes cermentový trimr (má být přibližně 4,5 V). [6]

Pro správnou komunikaci mezi procesorem a zobrazovacím modulem je opět třeba zajistit převod napěťových úrovní. Je zde tedy využit druhý jednosměrný budič z obvodu 74HCT244 pro převod úrovní na řídicích signálech RS, R/W a E. Pro převod úrovní na paralelní datové sběrnici je použit obvod 74HCT255, což je osmibitový obousměrný neinvertující budič. Vstup budiče DIR pro řízení směru přenosu signálů je zapojen na výstup řídicího procesoru tak, aby si procesor sám mohl plně řídit komunikaci se zobrazovacím modulem.



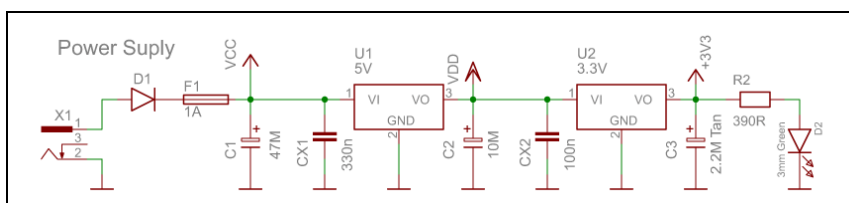
Obrázek 2.7 - Zapojení zobrazovacího modulu

Podsvěcovacím polem LCD displeje je serio-paralelní zapojení 2x28 LED diod. Při rozsvícení na něm vzniká úbytek napětí 4,2 V. Z modulu je vyvedena společná katoda a společná anoda podsvěcovacího pole. V návrhu jsou tyto vývody zapojeny tak, aby mohl řídicí procesor vypínat a zapínat podsvětlení dle aktuální potřeby (podsvícení při ovládání uživatelem, dle denní doby, apod.). Pro napájení podsvícení je zvolen typický proud uváděný výrobcem 280 mA [6]. Pro jeho dosažení je pole připojeno na 12V zdroj přes předřadník s odporem 27 Ω . Ovládání pole výstupem z procesoru je kvůli velkému proudovému zesílení realizováno pomocí dvou komplementárních spínacích stupňů.

2.3.5 Napájecí zdroj

V konečném návrhu tedy vznikla potřeba třech různých napájecích napětí (napájecí zdroj elektroventilů není řešen v rámci řídicí jednotky). Řídicí procesorový modul potřebuje stabilizované napájení 3,3 V. Řadič LCD displeje a vstupní obvody čidel vlhkosti vyžadují 5V napájení a výstupní relátka mají jmenovité vstupní napětí 12 V. Pro napájení podsvícení LCD displeje je také použit 12V zdroj.

Jednotka je tedy navržena na napájení z 12V DC napájecího modulu a nižší napájecí napětí jsou pak odvozena pomocí integrovaných stabilizátorů napětí v kaskádním uspořádání. Zapojení zdroje dále obsahuje doporučené filtrační kondenzátory [6]. Na pravé straně zapojení napájecích stupňů je ještě vidět kontrolka napájení jednotky.



Obrázek 2.8 - Napájecí zdroj

2.3.6 Shrnutí

Zařízení řešené dle tohoto návrhu elektroniky umožňuje:

- připojení libovolných solenoidních elektroventilů do napájení 250V/10A AC a zároveň použití různých ventilů pro každý ze čtyř výstupů,
- nezávislé spínání jednotlivých výstupů a to i sepnutí více výstupů najednou na základě reálného času a aktuální vlhkosti ovzduší (podle obsaženého programu),
- připojení libovolných čidel vlhkosti s lineárním 5V výstupem, možnost použití různých čidel pro každý okruh,
- poměrně přesné měření vlhkosti ovzduší čtyřmi nezávislými čidly na různých místech (jedno pro každý okruh), vzorkování vlhkostí v přesně daných časových okamžicích (podle obsaženého programu a nastavení uživatele),
- ukládání historie naměřených hodnot,
- díky kontrolkám na předním panelu je okamžitě vidět, zda je zařízení v provozu a zda je sepnutý některý z výstupů,
- použití velkého LCD displeje k zobrazení údajů o aktuálním stavu i budoucích stavech zavlažovacího systému a použití podsvícení LCD displeje podle aktuální potřeby, nebo i k indikaci různých událostí (dle obsaženého programu a zvoleného nastavení),
- vzdálené ovládání přes Ethernetovou přípojku i přímé lokální ovládání přes textové menu realizované pomocí čtyř směrových tlačítek a alfanumerického LCD displeje,
- vytváření a správu uživatelských programů pro ovládání zavlažování,
- uchování uložených dat a nastavení a správného reálného času i při dlouhodobém výpadku napájení.

Po konstrukci prototypu řídicí jednotky byly zjištěny následující nedostatky:

- Většina obvodů ovládaných centrálním procesorem je připojena na jeho obousměrné piny, po zapnutí zařízení (připojení napájení) jsou tyto signály na krátký okamžik v nedefinovaných stavech.

Současné řešení: V programu procesoru je konfigurace portů umístěna hned jako první akce po zapnutí zařízení, ovládané periferní obvody jsou tedy v nedefinovaných stavech jen velmi krátce a uživatel žádné neobvyklé chování ani nepostřehne.

Ideální řešení: Bylo by vhodné vypracovat novou verzi desek plošných spojů, doplněných o PullUp/PullDowny na příslušných signálech.

- Na 5V stabilizátoru 7805 dochází k relativně velkému skoku napětí (6,6 V) a může jím procházet proud až několik stovek mA spotřebovávány zejména centrálním řídicím modulem. Tato součástka je, dle svého datasheetu, na takovéto výkony sice dimenzována, co bylo však v současném návrhu mírně podceněno je její nadměrné zahřívání.

Současné řešení: Součástka je připevněna na velký pasivní chladič, procházející proud je možné omezit snížením taktovací frekvence centrálního procesoru popřípadě vypnutím Ethernetového řadiče.

Ideální řešení: Optimalizace zapojení kaskády napájecího obvodu a přepracování desek plošných spojů.

- Na základní desce zařízení nejsou provedeny ochranné obvody pro výstupní okruhy spínacích relé (ochrany proti přepětí, nadproudu, apod.) ani ochranné či filtrační obvody pro připojená čidla.

Současné řešení: Vzhledem k předpokladu, že řídicí jednotku bude zapojovat kvalifikovaná osoba (při celkové realizaci či renovaci zavlažovacího systému), by tato osoba měla doplnit zapojení o výše zmíněné obvody, dle konkrétního provedení obvodů ventilů a čidel. Menší provedení doplňujících obvodů by se mělo vejít i dovnitř krabičky jednotky ke stávající elektronice.

Ideální řešení: Přepracovat současnou verzi desek plošných spojů a doplnit elektroniku o výše zmíněné obvody v univerzálním či nastavitelném provedení.

2.4 Konstrukční řešení

Při návrhu konstrukčního řešení, tedy desek plošných spojů (dále jen DPS), bylo dbáno zejména na kompaktnost celého výrobku a na účelové rozmístění součástek (konektory, kontrolky, apod.). Kvůli zjednodušení layoutu plošných spojů bylo provedeno několik drobných úprav schématu elektrického zapojení (výměna některých součástek, přepojení určitých signálů, apod.)

Tato kapitola popisuje postup návrhu fyzického rozvržení zařízení. Podkapitoly jsou řazeny chronologicky tak, jak postupoval návrh konstrukčního řešení. V závěru kapitoly jsou shrnuty výsledné mechanické parametry jednotky a také technické nedostatky zjištěné po skutečné realizaci zařízení.

2.4.1 Třívrstvá struktura

Jedním z výchozích bodů pro návrh layoutu součástek na DPS byl fakt, že v zařízení jsou použity dva již hotové, relativně velké moduly, modul řídicí (s procesorem Rabbit) a modul zobrazovací (s LCD displejem). A oba tyto moduly předpokládají připojení a připevnění na jinou (hlavní) desku. U zobrazovacího modulu je jednoznačně nutné umístění navrchu finálního zařízení tak, aby byla viditelná celá plocha LCD displeje, kdežto u řídicího modulu postačuje dostupnost konektoru RJ-45 (z boku zařízení) a popřípadě programovacího konektoru. Takto vznikla úvodní myšlenka umístit tyto dva moduly nad sebe pomocí dlouhých kolíkových konektorů a distančních sloupků, vzhledem k tomu, že umístění modulů vedle sebe by podstatně zvětšilo plochu základní desky a současně by vytvářelo mnoho nevyužitelného prostoru pro umístění ostatních součástek.

Při umístění většího LCD modulu nad řídicí modul s menší plochou vzniká určitý prostor s velkým odstupem mezi deskami spojů (mezi základní a zobrazovací deskou), do kterého se na výšku vejde jakákoliv součástka z návrhu zařízení. Do prostoru pod řídicím RCM modulem pak mohou být umístěny alespoň ploché součástky jako jsou integrované obvody v pouzdrech DIP (při zachování odstavu 2 mm od spodní strany řídicí desky, který doporučuje Rabbit [7]). Prostor na základní desce tak může být lépe využit a nadále se zmenšuje celková plocha zařízení. Další redukce plošných spojů pak bylo možné dosáhnout ještě použitím plochých vícežilových kabelů pro spoje mezi deskami namísto přímých kolíkových konektorů.

2.4.2 Účelové rozmístění součástek

Tak jako bylo důležité umístění zobrazovacího modulu nad modul řídicí a ne naopak, bylo důležité i vhodné umístění kontrolky a ovládacích tlačítek, nejlépe i ve vztahu k samotnému LCD displeji. Proto bylo od počátku návrhu DPS počítáno s výrobou samostatné desky pro ovládací a signalizační prvky (Interface Board), která bude společně se zobrazovacím modulem umístěna navrchu zařízení.

Také bylo důležité umístění všech konektorů včetně řídicího modulu s konektorem RJ-45 na kraj hlavní (spodní) desky, aby bylo možné pohodlně připojit všechny přírady z boku zařízení. V další řadě pak šlo i o rozumné umístění již méně používaných účelových prvků jako je proudová pojistka, programovací konektor, tlačítko Reset a trimr pro seřízení napájecího napětí LCD displeje.

2.4.3 Finální forma

Původní návrh layoutu počítal s již výše zmíněným umístěním procesorového modulu pod zobrazovacím modulem. Pod LCD modulem měl být umístěn i kompletní vstupní obvod čidel vlhkosti s konektory vyvedenými na pravou stranu zařízení a obvod napájecího zdroje se vstupním DC konektorem vyvedeným na spodní stranu společně s konektorem z modulu RCM pro Ethernetový kabel. Nalevo od LCD modulu pak měly být ve spodní vrstvě umístěny výstupní relátka (s výstupní svorkovnicí na levém kraji základní desky) a v horní vrstvě ovládací tlačítka a kontrolky.

Při tomto rozmístění součástek však základní deska (a tudíž i celé zařízení) dostávala značně obdélníkový formát, na který je mimo jiné i nelehké sehnat konstrukční krabici. Proto bylo rozvržení součástek přepracováno na více čtvercovou verzi, kde jsou tlačítka a kontrolky umístěny na spodní straně od displeje. Všechny konektory a účelové součástky jsou umístěny na bocích zařízení, také všechny výkonové součástky jsou na jednom kraji základní desky a lze tak šikově namontovat chlazení. Navíc tato varianta vedla k celkově kompaktnějšímu zařízení a elegantnějšímu layoutu plošných spojů.

Celá elektronika je umístěna v plastové konstrukční krabici U-KP06, jedná se o variantu bez větracích otvorů. Tato krabice byla vybrána jako nejvhodnější z nabídky GM Electronic. Krabice je o něco málo větší, než bylo na dané rozměry bloku elektroniky potřeba. Z toho důvodu jsou mimo jiné umístěny i chladiče výkonových součástek celé uvnitř konstrukční krabice, oproti původnímu návrhu, kdy bylo počítáno s vyčníváním chladičů ven z krabice. Tato nová varianta vypadá elegantněji.

Blok elektroniky je v krabici upevněn pomocí 4 šroubů procházejících podlázkou krabice, všechny díly krabice jsou pak pevně staženy pomocí dalších čtyř šroubů. Na vrchní straně krabice jsou vyhotoveny otvory pro LCD displej, pro tlačítka a kontrolky. Na straně přední jsou otvory pro konektor napájení a pro připojení Ethernetového kabelu. Na levém boku krabice, u chladičů výkonových součástek, jsou větrací otvory. Na zadní straně zařízení, popřípadě i na pravém boku, se počítá s vyvedením kabelů k ovládacím elektroventilům a kabelů od vlhkostních čidel. Vzhledem k nejasnosti konečné podoby těchto vedení nejsou příslušné průchodky v krabici dosud provedeny.

2.4.4 Shrnutí

- Výsledné zařízení má podobu jediné kompaktní krabice, je relativně malé, vhodné pro umístění na vodorovnou plochu či zavěšení na zeď.
- Na vrchní straně zařízení je pouze lokální ovládací rozhraní a kontrolky v jednoduchém a přehledném uspořádání, konektory jsou umístěny na přední straně krabice. S průchodkami pro vedení k ventilům a čidlům se počítá na zadní straně (popřípadě i na boku).
- Krabici lze snadno rozmontovat vyšroubováním 4 šroubů. Jejich hlavičky ústí na spodní straně krabice, nejsou tedy normálně vidět a nenarušují jednoduchý design jednotky.
- *Výsledné rozměry sestaveného bloku elektroniky:*
110 × 97 × 42 mm (šířka × výška × hloubka)
- *Vnější rozměry použité konstrukční krabice:*
150 × 130 × 50 mm (šířka × výška × hloubka)

Protože krabice řídicí jednotky byla vybírána až v poslední fázi návrhu zařízení, nesouladí zcela ideálně s konstrukcí elektroniky:

- Konektory na přední straně bloku elektroniky nejsou přímo u stěny krabice (jsou trochu „hlouběji“), před jedním z konektorů pro připojení čidel je poměrně nešikovně umístěn vnitřní sloupek krabice.

Ideální řešení: Přepřeprogramování desek plošných spojů pro vhodné umístění do této konkrétní krabice (mírné zvětšení, posunutí konektorů, přidání výřezů pro sloupky krabice).

- Kontrolky Ethernetové komunikace na řídicím modulu. (v blízkosti konektoru RJ-45) jsou z výše uvedeného důvodu také „hlouběji“ v krabici, navíc jsou v porovnání s ostatními ovládacími a zobrazovacími prvky jednotky poměrně drobné a nevýrazné. Použití těchto originálních ledek řídicího modulu a „okénka“ u konektoru RJ-45 se tedy neukázalo jako příliš dobré či estetické řešení.

Současné řešení: „Okénko“ pro kontrolky není vyhotoveno. Avšak za předpokladu, že řídicí jednotka bude pravděpodobně umístěna v blízkosti síťového směšovače či ovládacího PC, funkčnost komunikace lze snadno zkontrolovat na připojeném zařízení.

Ideální řešení: Nahrazení původních kontrol modulu vlastními kontrolkami umístěnými na předním panelu jednotky u ostatních již navržených kontrol.

3 NÁVRH ŘÍDICÍHO PROGRAMU JEDNOTKY

Tato část práce stručně popisuje postup návrhu a konečnou podobu programového vybavení řídicí jednotky. První podkapitola shrnuje počáteční návrh obecné struktury a kostry řídicího programu, je zde podrobně vysvětleno, jaká byla zvolena koncepce programu a proč došlo k její změně (kapitola **Paralel Port F Bug**). Druhá podkapitola podrobně popisuje činnost hlavních procesů programu. Podkapitoly 3 a 4 vysvětlují podobu a vnitřní funkci ovládání řídicí jednotky. Na konci kapitoly je uvedeno shrnutí funkcí řídicí jednotky umožňovaných současnou verzí programu a také shrnutí funkcí, které program zatím neumožňuje, ale bylo by možné je v budoucnu doplnit.

V této kapitole jsou uváděny úryvky ze zdrojového kódu řídicího programu. Tyto části kódu byly upraveny (zkráceny) pro lepší názornost ukázek. Symbol „...“ nahrazuje rozsáhlejší nebo nesouvisející části kódu. Program byl vytvořen ve vývojovém prostředí *Dynamic C*[®] 9.62 a celkem čítá přes 6600 řádků.

3.1 Návrh základní struktury programu

Struktura řídicího programu zařízení vychází hlavně z faktu, že jednotka má dvě nezávislá ovládací rozhraní (lokální ovládané směrovými tlačítky a vzdálené ovládané přes Ethernetovou přípojku) a že je současně potřeba vykonávat úlohy naplánované na dané okamžiky v reálném čase (změna stavu připojených ventilů, změření a uložení aktuálních vlhkostí ovzduší, apod.) nezávisle na stavu ovládacích rozhraní.

Program je proto rozdělen na 3 hlavní procesy: *HTTPHandler* realizující vzdálené ovládací rozhraní, *ButtonsHandler* realizující lokální ovládací rozhraní a *TaskHandler* spouštějící jednotlivé naplánované úlohy. (To jsou hlavní úlohy jednotlivých handlerů, detailnější popis jejich skutečné činnosti bude uveden v kapitole **Hlavní procesy**.)

3.1.1 Uspořádání hlavních procesů

Za začátku návrhu byla snaha zpracovat většinu programového kódu jako reakce na přerušení procesoru: Ethernetový řadič v přerušovacím režimu, ovládací tlačítka připojená na piny externích přerušení, spouštění naplánovaných úloh jako reakce na přerušení časovačů. Základní prioritní pásmo (kdy procesor nevykonává žádnou obsluhu) programu by tak bylo využito jen minimálně a bylo by tak možné i uspaní procesoru ve chvílích, kdy není nutná žádná činnost.

Jak se však postupně ukázalo, při takovéto koncepci programu by bylo značně obtížné vyhnout se možným konfliktům jednotlivých procesů. Navíc aplikační rozhraní realizující funkci HTTP Serveru předpokládá periodické volání funkce *http_handler* právě v základním prioritním pásmu (Priorita „přerušeni“ 0). Vzhledem k tomu, že se jedná o nejsložitější a časově nejnáročnější proces v programu, uspávání procesoru také není na místě.

Žádný ze tří hlavních procesů však ve skutečnosti nevyžaduje kritické časování: kontrolu a spouštění naplánovaných úloh stačí provádět jen jednou za každou celou sekundu, na stisk tlačítka stačí reagovat v řádu desítek až stovek milisekund a funkci *HTTPHandleru* je vzhledem k relativně velkému bufferu Ethernetového řadiče optimální volat asi 10krát za sekundu [8]. Proto bylo nakonec zvoleno umístění všech tří procesů do hlavního prioritního pásma (priorita 0) časově „za sebou“ v nekonečné smyčce (struktura „BigLoop“ – viz **Obrázek 3.1**) a v pásmu přerušeni byly ponechány jen drobné části kódu s nutností přesnějšího časování - viz kapitola **Výsledné časování**.

```

/***** MAIN PROGRAM *****/

void main() {

//===== Initialization section =====//

...

//turn off BL to indicate Init-Done
blEnable = 0;

//===== Normal operation =====//
while(1) { //infinite Big Loop

    //handle Planned events (RT Daemon)
    taskHandler();

    //handle Local Control Interface (Buttons Daemon)
    buttonsHandler();

    //handle Remote Control Interface(HTTP Daemon)
    http_handler();

} //End Big Loop

} //End Main

```

Obrázek 3.1 - Hlavní kód programu

Takto je v daném okamžiku vykonáván vždy jen kód jediného procesu a další proces přichází na řadu až po dokončení celé základní operace předchozího procesu (tzv. jeden „tick“). Procesy se tak vzájemně nepřerušují, celá aplikace se významně zjednodušuje a předcházení konfliktům je podstatně snazší.

3.1.2 Výsledné časování

Pro časování filtru zákmytů ovládacích tlačítek je využito časovače procesoru Timer A. Časovač je nastaven pro vyvolání přerušení každou 1 ms, při všech programem dovolených nastavení taktovací frekvence (přesnost časování však mírně klesá současně s taktovací frekvencí).

Timer B byl rezervován pro časování komunikace s modulem LCD displeje. Například vykonání instrukce výmazu celého displeje trvá 1,64 ms [5], místo čekání na display by procesor mohl využít tento čas k vykonávání jiné části kódu programu. V současné verzi programu je však Timer B využit pouze k regulaci jasu podsvícení displeje pomocí PWM (důvod viz kapitola [Paralel Port F Bug](#)). Četnost přerušení od toho časovače zatím není korigována vzhledem ke změně taktovací frekvence procesoru, v důsledku toho je při nastavení frekvence menší než 11 MHz patrné blikání displeje.

Obslužné rutiny přerušení časovačů trvají řádově několik stovek hodinových cyklů. Doba trvání ticku (provedení základní části procesu v hlavní smyčce) *ButtonsHandleru* je také zanedbatelná (méně než 0,5 ms), jeden tick *TaskHandleru* trvá maximálně 3 ms a nejdelší tick *HTTPHandleru* trvá necelých 45 ms. Uvedené časy platí pro taktovací frekvenci 22,1 MHz.

3.1.3 Koncepce ovládacích rozhraní

Protože řídicí jednotka má dvě nezávislá ovládací rozhraní, bylo třeba vyřešit (či předejít) jejich případné konflikty. Zejména kvůli vyhnutí se situaci, kdy by si dva uživatelé vzájemně měnili nastavení jednotky „pod rukama“, bylo zvoleno řešení, kdy při nastavování řídicí jednotky pomocí jednoho ovládacího rozhraní, je to druhé kompletně zablokováno. Pro zamezení neúmyslnému trvalému zablokování jednoho z ovládacích rozhraní byly přidány ještě timeouty pro odhlášení uživatele při neaktivitě na daném rozhraní (odblokování druhého rozhraní).

Způsob provedení lokálního ovládání byl vzhledem k technickým možnostem lokálního rozhraní (4 ovládací tlačítka, čtyř-řádkového LCD displej) předem poměrně jasný. Ovládání mělo být realizováno jako textové menu, kdy se na LCD displeji zobrazují informace o stavu zavlažovacího systému a možná nastavení. A pomocí ovládacích tlačítek lze měnit hodnoty těchto nastavení a přecházet mezi jednotlivými obrazovkami menu.

Pro vzdálené (webové) ovládání bylo již více možností realizace a vybrání vhodného řešení bylo o něco náročnější. Nakonec bylo pro svou jednoduchost a všestrannou výhodnost zvoleno ovládání ve stylu virtuálního lokální rozhraní - ovládání, které vytváří dojem, jako by uživatel ovládal jednotku přímo přes její lokální rozhraní. Tato varianta ovládání poskytuje přesně stejné funkce jako ovládání lokální, uživatel jednotky se nemusí učit pracovat se dvěma různými ovládacími rozhraními, navíc každá změna v programovém kódu menu (viz kapitola **Ovládací Menu jednotky**) se automaticky projeví shodně na obou ovládacích rozhraních. Velkým přínosem této koncepce je také malá velikost programového kódu i HTML stránek webového rozhraní a tudíž ušetření volného místa paměti.

3.1.4 Paralel Port F Bug

Při ožiování LCD displeje byla zjištěna závažná chyba – při pokusech o komunikaci se zobrazovacím modulem byla na datové sběrnici (Port A) v kritický okamžik vždy vystavena chybná data, zcela neodpovídající datům posílaným procesorem. Při bližším zkoumání bylo zjištěno, že správná data se na sběrnici sice vystaví, ale po vykonání další instrukce – změně řídicího signálu displeje připojeného k portu F, jsou na datové sběrnici opět chybná data. Tento jev se vždy naprosto stejně opakoval. Podařilo se mi dohledat, že tato chyba je dobře známým interním bugem daného procesoru:

„However if Parallel Port A is used as an output or is used as the bidirectional bus of the slave port, then writing to any of the Parallel Port F registers will cause a spurious write to the Parallel Port A register.“ [10]

Protože zprovoznění LCD displeje by tím pádem vyžadovala změnu obvodového zapojení a předělání v tu dobu již kompletně osazené desky, soustředil jsem se na dokončení ovládání přes Ethernetovou přípojku, které bylo v původním zadání práce.

Vzhledem ke zvolené koncepci ovládacích rozhraní (viz kapitola **Koncepce ovládacích rozhraní**) byl v této fázi realizace kód pro lokální ovládání již připraven a v současné verzi programu je v podstatě i téměř dokončen. Nicméně bez funkčního displeje ovládání pomocí textového menu postrádá smysl, ovládací tlačítka a podsvícení displeje jsou přesto plně funkční. Pro využití ovládacích tlačítek jim byla přiřazena alespoň funkce manuálního ovládání a podsvícení bylo využito k indikaci událostí dle nastavení v menu Display Settings.

Vzhledem k tomu že „nastavování“ pomocí lokálního ovládání takto znamená pouze reakce na stisk tlačítka (zápis do paměti a změna stavu výstupů) netrvající déle než 0,5 ms je v současném programu prováděno pouze blokování lokálního rozhraní při přihlášení ke vzdálenému rozhraní a ne naopak.

3.2 Hlavní procesy

V této kapitole jsou podrobněji vysvětleny jednotlivé činnosti prováděné hlavními procesy programu, které byly představeny v kapitole [Návrh základní struktury programu](#). Jsou také uvedeny souvislosti mezi těmito procesy a dalšími částmi řídicího programu.

3.2.1 Buttons Handler

Tento proces při každém ticku (jednom jeho volání z hlavní smyčky programu) kontroluje stav ovládacích tlačítek. Pokud je stav stejný jako v předchozím ticku, žádná akce se neprovádí a přechází se k dalšímu procesu ve smyčce. Pokud dojde ke změně stavů tlačítek (stisk/puštění alespoň jednoho z tlačítek), zjišťuje se k jaké konkrétně změně došlo a popřípadě se volá příslušná akce. Dále se spouští časovač (Timer A) pro odfiltrování zákmytů tlačítka, čímž se přejde do režimu „Waitout“.

```
//===== Buttons Handler - Main Routine =====//
void buttonsHandler() {

    char actBtStates;

    if(notInWaitout()) {
        //do nothing in WaitOut period

        //check fot Bt States change
        actBtStates = getBtStates();
        if(actBtStates != prevBtStates) {

            //Check New State
            switch(actBtStates) {

                //release - do nothing
                case BT_NONE: break;

                //Direction Buttons
                case BT_LEFT:
                    if(prevBtStates == BT_NONE) //single bt press
                    if(interface == NONE){ //not blocked by TCP/IP
                        buttonsBl();
                        buttonsOverride(OUT_1); //perform action
                    } break;

                ...

            } //State case

            //Change to Waitout mode
            startWaitout();
            prevBtStates = actBtStates;

        } //Change
    } //Waitout
} //End Buttons Hanlder
```

Obrázek 3.2 - Buttons Handler

V režimu „Waitout“ neprovádí *ButtonsHandler* žádnou kontrolu ani akce – změny stavů tlačítek jsou v tomto období ignorovány. Po uplynutí přibližně 10 ms časovač A obnoví normální režim a sám sebe zastaví. *ButtonsHandler* pak opět pokračuje v kontrolách stavu tlačítek.

Co se týče volaných akcí při stisku tlačítek – původní koncepce programu počítala se zablokováním vzdáleného ovládacího rozhraní a s voláním Menu „Handleru“ (jak bylo uvedeno v kapitole **Koncepce ovládacích rozhraní**). V současné verzi je volána pouze funkce *buttonsOverride*, tedy úprava manuálního ovládání, a pokud je tak nastaveno i indikace podsvícením LCD displeje (funkce *buttonsBl*).

3.2.2 HTTP Handler

Tento proces realizuje funkcionalitu samotného HTTP Serveru. Zde je využito standardního aplikačního rozhraní knihoven „dcrtcp.lib“ a „http.lib“. Při spuštění programu se inicializuje rozhraní TCP/IP a HTTP Daemon. Během této inicializace je nakonfigurován Ethernetový řadič, nastavena implicitní IP adresa 10.10.6.100 a naslouchání na portu 80 pro HTTP Demon. Celý tento proces trvá přes 3 s (při 22,1 MHz). Proto je po zapnutí řídicí jednotky, až do dokončení inicializace, rozsvíceno podsvícení displeje k indikaci probíhající inicializace zařízení.

```
//===== HTTP StartUp =====  
void StartUpHTTP() {  
  
    //initiate the TCP/IP layer  
    sock_init();  
  
    //initialize HTTP Daemon  
    http_init();  
  
    //set port 80 as reserved port  
    tcp_reserveport(80);  
  
} //end HTTP StartUp
```

Obrázek 3.3 - Inicializace HTTP Serveru

Tick *HTTPHandler* je pak realizován voláním standardní knihovní funkce *http_handler*. Ta se stará o navázání spojení s klientem a zasílání odpovídajících dat. Dále pak volá *shtml_handler* (při odesílání souborů *.shtml) pro realizaci SSI, spouští CGI funkce programu (při požadavcích na „soubory“ *.cgi) a další. Další popis realizace webového rozhraní je uveden v kapitole **Webové ovládací rozhraní**.

3.2.3 Task Handler

Hlavní funkcí *TaskHandleru* je spouštění naplánovaných úloh: změna stavu výstupů a měření vlhkosti ovzduší. Handler ovšem zajišťuje ještě funkci timeoutů ovládacích rozhraní a podsvícení LCD displeje (pro dočasné podsvícení) a také refresh zobrazení displeje pokud jsou právě zobrazována proměnná data (aktuální čas, aktuální vlhkost ovzduší, apod.).*

Většina akcí prováděných *TaskHandlerem* je určena pro spouštění právě jednou za vteřinu. Handler tedy v každém svém ticku nejprve kontroluje, zda se změnil čas virtuálního timeru SEC_TIMER (zda uplynula celá 1 vteřina), a až potom spouští dané akce. Pokud se čas nezměnil, handler žádné akce nespouští (s výjimkou případného měření vzorků - viz dále) a následuje další proces v hlavní smyčce programu. Dále jsou popsány jednotlivé činnosti *TaskHandleru*, na příslušných obrázcích jsou uvedeny vždy jen části kódu související s danou činností.

* V současné verzi programu TaskHandler z důvodů popsaných v kapitole 3.1.4 Paralel Port F Bug nespravuje timeout přihlášení k lokálnímu ovládacímu rozhraní a funkce obnovení zobrazení displeje je zatím určena pouze pro webové ovládací rozhraní.

Timeouty: Pokud běží časovač timeoutu (hodnota příslušné proměnné *Expire* je nenulová), *TaskHandler* provádí odpočet. Pokud čas timeoutu vyprší (hodnota je snížena až na nulu), je provedena příslušná akce: zhasnutí podsvícení nebo odhlášení uživatele na daném ovládacím rozhraní (a odblokování druhého ovládacího rozhraní).

```
//===== Task Handler =====//
void taskHandler(){

    ...

    //----- Events Section -----//
    //Check Sec rollover
    if(actTime != SEC_TIMER) {
        actTime = SEC_TIMER; //refresh time

        //Connection Timeout
        if(remoteExpire != 0) {
            remoteExpire--; //Timeout Tick

            if(remoteExpire == 0) { //Timeout Occured
                interface = NONE; //Log Out
                if(blMode == BL_CONNECT) //Indicate Unblocked
                    blEnable = 0;
            } //Timed-out
        } //End ConnTimeout

        //BackLight Timeout
        if(blExpire != 0) {
            blExpire--; //Timeout Tick

            if(blExpire == 0) { //Timeout Occured
                blEnable = 0; //Turn Off BackLight
            } //Timed-out
        } //End ConnTimeout

        ...

    } //Sec rollover
    //----- End EventsSection -----//
} //End Task Handler
```

Obrázek 3.4 - Task Handler - Správa timeoutů

Refresh zobrazení: Pokud je na displeji právě zobrazen jeden z ViewScreenů (viz kapitola **Koncepce ovládacího menu**), je každou uplynulou vteřinu obnoven zobrazovaný aktuální čas a zobrazovaná data dané obrazovky: aktuální program zavlažování, nejbližší změna stavu zavlažovacího systému, vlhkosti ovzduší aktuálně měřené připojenými čidly nebo stav manuálního ovládání.

```
//===== Task Handler =====
void taskHandler(){

    ...

    //----- Events Section -----//
    //Check Sec rollover
    if(actTime != SEC_TIMER) {
        actTime = SEC_TIMER; //refresh time

        ...

        //Display Refresh
        dispRefresh();
        //(for LCD Disp Refresh
        //& Browser Refresh)

    } //Sec rollover
    //---- End EventsSection ----//

} //End Task Handler

...

/* Dislay Refresh: Called from Events Handler */
void dispRefresh() {

    //View Screens
    if(menuState < 10) { //View Screens
        //Refresh displayed time
        mktm(&tmpTm, SEC_TIMER);
        writeDateTimeSecs(displine[0], &tmpTm);
    } //End View Screens

    //Program Screen
    if(menuState == 1)
        refreshProgramScr();

    ...

    //Manual Override Screen
    else if(menuState == 4)
        refreshOverrideScr();

} // End Display Refresh
```

Obrázek 3.5 - Task Handler - Refresh zobrazení

Měření vlhkosti ovzduší: Každou celou vteřinu kontroluje *TaskHandler*, zda není naplánováno měření vlhkosti. Pokud ne, kontroluje ještě, zda je naplánovaný okamžik příštího měření aktuální a pokud není (je v minulosti), pak naplánuje nový. Pokud nastane okamžik naplánovaného měření vlhkosti handler si nastaví počet vzorků k měření podle toho, z kolika vzorů má být výsledná hodnota zprůměrována (nastavení dostupné v menu Global Measure Settings).

V následujícím ticku *TaskHandler* zjistí, že má změřit vzorek vlhkostí, vzorek naměří a sníží počet vzorků ke změření. Tento krok se pak opakuje dokud není naměřen celý počet naplánovaných vzorků. Po změření posledního vzorku (snížení počtu vzorků k měření na nulu) se z naměřených hodnot vypočte průměrná hodnota a je uložena do historie naměřených vlhkostí. Tato měření a uložení výsledné hodnoty se provedou bez ohledu na to, zda došlo ke změně reálného času (uplynutí celé 1 s) tedy se zpožděním vždy jednotek až desítek milisekund od předchozího ticku.

```
//===== Task Handler =====
void taskHandler(){

    //----- Measure Section -----//
    if(pendingMeasures > 0) {

        //perform measures
        if(sensorConnected(1)) m1 += measureAndConvert(1);
        if(sensorConnected(2)) m2 += measureAndConvert(2);
        if(sensorConnected(3)) m3 += measureAndConvert(3);
        if(sensorConnected(4)) m4 += measureAndConvert(4);

        //count down measures & check finished
        if(--pendingMeasures == 0) {
            //count averages & store to History
            ...
            //reset vals
            m1 = 0; m2 = 0; m3 = 0; m4 = 0;
        } //End finished if

    } //End Measure

    //----- Events Section -----//
    //Check Sec rollover
    if(actTime != SEC_TIMER) {
        actTime = SEC_TIMER; //refresh time

        ...

        //Measure Event
        if(measureStartDate < actTime) { //missed
            //reschedule
            while(measureStartDate <= actTime)
                measureStartDate += measureInterval;

        } else if(measureStartDate == actTime) { //match
            //initiate measure cycle
            pendingMeasures = averaging;

        } //else nothing (wait)
        //End Measure Event

        ...

    } //Sec rollover
    //----- End EventsSection -----//
} //End Task Handler
```

Obrázek 3.6 - Task Handler - Měření vlhkostí

Změna stavu výstupů: Každou celou vteřinu *TaskHandler* kontroluje, zda nenastala zavlažovacím programem naplánovaná změna stavu výstupů. Pokud ano, je uložen nový stav výstupů do proměnné *progOutput* a naplánována další změna stavu systému. Pokud není aktivní žádný zavlažovací program, do proměnné *progOutput* se ukládá nula (všechny výstupy vypnuty).

Pokud je aktivní manuální (nadřazené) ovládání, handler kontroluje jeho dobu platnosti a pokud je již neplatné, deaktivuje ho. Nakonec jsou stavy výstupů nastaveny v závislosti na aktuálním režimu ovládání (manuální nebo program). Díky tomuto provedení kódu lze obnovit stav výstupů dle aktuálního zavlažovacího programu (proměnná *progOutput*) po vypršení platnosti nadřazeného manuálního ovládání.

```
//===== Task Handler =====
void taskHandler(){

    ...

    //----- Events Section -----//
    //Check Sec rollover
    if(actTime != SEC_TIMER) {
        actTime = SEC_TIMER; //refresh time

        ...

    //----- Program Events -----
    // - - - Program Update - - - -

        //Program Inactive
        if(noActProg()) {
            progOutput = 0; //Reset Output

        //Active Program
        }else{
            ... //determine progOutput & next Step
        }//End Program Active If

    // - - - Override Update - - - -

        if(ovrIsActive())
            //check Due Date to turn Off
            if(SEC_TIMER >= override.dueDate)
                setOverride(OVR_INACTIVE,0);

    // - - - Outputs Update - - - -

        if(ovrIsActive())
            setOutputs(override.state);
        else setOutputs(progOutput);

    //-----

    ...

    } //Sec rollover
    //---- End EventsSection ----//

} //End Task Handler
```

Obrázek 3.7 - Task Handler - Aktualizace výstupů

3.3 Webové ovládací rozhraní

Vzdálené ovládací rozhraní v současné verzi programu sestává ze tří (nebo čtyř, podle úhlu pohledu) webových stránek. Hlavní stránka rozhraní je realizována pomocí SSI (výstup informací k uživateli), provádění požadovaných akcí na straně řídicí jednotky je realizováno pomocí CGI funkcí (vstup příkazů od uživatele), po provedení akce je uživatel vždy přesměrován na další stránku.*

Ukázka řídicí jednotkou vygenerovaných stránek je na přiloženém CD. Jedná se ale pouze o ukázkou, stránky jako takové bez řídicí jednotky postrádají funkcionalitu. Ukázka podoby webového rozhraní je také v příloze [Webové ovládací rozhraní](#). Obrázky byly mírně upraveny pro lepší čitelnost.

Dále jsou postupně popsány jednotlivé HTML stránky rozhraní – jejich účel a provedení a následně činnosti prováděné volanými CGI funkcemi.

3.3.1 HTML Stránky

index.shtml: Toto je hlavní stránka ovládacího rozhraní, je implicitně zobrazována při zadání IP adresy zařízení do webového prohlížeče. Její zdrojový kód ve skutečnosti obsahuje pouze hlavičku stránky, zbytek je generován pomocí SSI (příkaz „`<!--#exec cmd="/CheckState.cgi"-->`“). Díky této koncepci se není možné do ovládacího rozhraní „nabourat“ například zadáním přímo hlavní ovládací stránky do prohlížeče – uživatel se musí korektně přihlásit.†

Pokud uživatel ještě není přihlášen, je sestavena a poslána stránka přihlášení (viz stránka `login.html` na CD nebo příloha [C.1](#)). Po kliknutí na odkaz „Log In“ na této stránce je volána funkce `cgiLogin`.

* Provedení a design webového rozhraní byly inspirovány ukázkovými programy k vývojovému prostředí Dynamic C® 9.62, bylo využito původního obrázku RABBIT1.GIF. Použité fonty byly čerpány z „[www.azfonts.net](#)“ a „[www.1001freefonts.com](#)“. Stránky byly odladěny v prohlížeči Google Chrome při rozlišení 1280x800.

† V původní koncepci programu, kde bylo počítáno i s blokováním webového rozhraní, měly stránky obsahovat i kontrolu, zda je v daném okamžiku vůbec možné se přihlásit. Pokud ne, pak by nesmělo být možné se k ovládání jakkoli dostat – proto důraz na zamezení možnosti „nabourání“ do systému.

Pokud je uživatel již přihlášen, funkce *cgiCheckState* sestaví hlavní ovládací stránku (viz stránka *main.html* na CD nebo příloha C.2). Na stránce je vygenerován aktuální obsah „displeje“ a aktuální stav kontrol výstupů. Obnovením stránky v prohlížeči se vygeneruje nová aktuální stránka a současně se vynuluje timeout přihlášení. Na stránce je odkaz „Log Out“ volající funkci *cgiLogout* a „tlačítka“ jednotky jsou odkazy volajícími jednotlivé funkce CGI tlačítek (např. *cgiBtLeft*).

logout.html: Na tuto stránku je uživatel přesměrován po korektním odhlášení. Předpokládá se, že po zobrazení této stránky již uživatel nebude pokračovat v ovládání řídicí jednotky. Nicméně je zde alespoň odkaz pro opětovné přihlášení (odkaz „Log In“ volající funkci *cgiLogin*).

errorr.html: Na tuto stránku je uživatel přesměrován v případě, že se pokouší o provedení nějaké akce (zpravidla stisknutím „tlačítka“ jednotky na hlavní ovládací stránce) bez platného přihlášení. K tomu dojde nejčastěji tehdy, když je uživatel během své neaktivity odhlášen řídicí jednotkou kvůli vypršení timeoutu přihlášení. V dalším případě se tak stane, pokud se uživatel například pokouší přímo spustit CGI funkci tlačítka zadáním jejího URL do prohlížeče, bez korektního přihlášení do webového rozhraní. Takto je opět zamezeno „nabourání“ do systému.

3.3.2 CGI Funkce

Všechny CGI funkce obsahují kód pro indikaci podsvícení LCD displeje (skutečného displeje na řídicí jednotce) dle aktuálního nastavení v menu Display Settings - funkce *actionBl* popřípadě kódy „*if(blMode == BL_CONNECT) ...*“.

cgiLogin: Tato funkce je volána při korektní žádosti o přihlášení ke vzdálenému ovládacímu rozhraní. Pokud uživatel ještě není přihlášen, funkce zablokuje lokální ovládací rozhraní (současně přejde do stavu přihlášený), spustí timeout přihlášení (délka timeout dle nastavení v menu Systém Settings) a inicializuje menu řídicí jednotky na nastavenou výchozí obrazovku. Dále je uživatel přesměrován zpět na index.shtml, kde je následně voláním *cgiCheckState* sestavena hlavní ovládací stránka.

```
// ==== Block local ctrl, Enable Remote ctrl ====//
int cgiLogin(HttpState* state){
    //if not logged in
    // -> perform login procedure & start timeout

    actionBl(); //BackLight Blink (if Enabled)

    if(interface == NONE) {

        interface = REMOTE;
        startConnTimeout();

        if(blMode == BL_CONNECT) //Indicate Blocked
            blEnable = 1;        //(if Enabled)

        menu(defScreenOvr()); //Initialize Menu
    } //End if not logged in

    //redirect to index in any case
    cgi_redirectto(state,REDIR_HOST);
    return 0;
} //End CGI Login
```

Obrázek 3.8 - CGI Funkce - Přihlášení

cgiLogout: Tato funkce je volána při správné žádosti o odhlášení ze systému. Pokud je uživatel přihlášen, funkce odblokuje lokální ovládací rozhraní (současně přejde do stavu odhlášený), zastaví timeout pro odhlášení a přesměruje uživatele na stránku logout.html.

```
// ==== Close remote ctrl, unblock local ctrl ====//
int cgiLogout(HttpState* state){
    //if logged in
    // -> perform logout procedure

    actionBl(); //BackLight Blink (if Enabled)

    if(interface == REMOTE) {

        interface = NONE;
        stopConnTimeout();

        if(blMode == BL_CONNECT) //Indicate Unblocked
            blEnable = 0;

    } //End logged in If

    //redirect to Logout.html in any case
    cgi_redirectto(state,REDIR_HOST "logout");
    return 0;
} //End CGI Logout
```

Obrázek 3.9 - CGI Funkce - Odhlášení

cgiCheckState: Je voláno SSI příkazem „`<!--#exec cmd=\"/CheckState.cgi\"-->`“ na stránce `index.shtml`. Pokud uživatel ještě není přihlášen, funkce odešle kód přihlašovací stránky. Pokud je přihlášen, funkce vynuluje timeout přihlášení a dynamicky sestaví a odešle kód hlavní ovládací stránky s aktuálním obsahem.

```
// ==== Check State: SSI Include on Index.shtml =====//
int cgiCheckState(HttpState* state){
    //if not logged -> send Login.html
    //if logged in -> send Main.html

    actionBl(); //BackLight Blink (if Enabled)

    if(interface == NONE) {
        //not Logged in -> Send Login page

        strcpy(ssiBuffer, LOGIN );
        strcat(ssiBuffer, PAGE_END );

    } else {
        //if Logged in -> Compose main page

        resetConnTimeout();

        strcpy(ssiBuffer, TABLE_HEADER );
        appendDisplay(ssiBuffer);
        strcat(ssiBuffer, BUTTONS );
        appendLeds(ssiBuffer);
        strcat(ssiBuffer, LOGOUT );
        strcat(ssiBuffer, PAGE_END );

    } //state if

    //Send
    cgi_sendstring(state, ssiBuffer);
    return 0;
} //End cgiCheckState
```

Obrázek 3.10 - CGI Funkce - Generování hlavní stránky

cgiBtCommon: Jedná se o funkci společnou pro všechna „tlačítka“ zařízení ve webovém rozhraní. Tato funkce nejprve zkontroluje, zda je uživatel korektně přihlášen. Pokud ne, neprovádí žádnou akci a přesměruje uživatele na stránku error.html. V opačném případě volá Menu „Handler“ s odpovídajícím příkazem (viz kapitola **Ovládací Menu jednotky**) a „přesměruje“ uživatele zpět na hlavní stránku (index.shtml) - tak se opět volá funkce *cgiCheckState*, která vynuluje timeout přihlášení a zobrazí nový aktuální obsah stránky.

```
// ==== CGI Buttons - performs actual actions =====//
int cgiBtCommon(HttpState* state, uint newState){

    actionBl(); //BackLight Blink (if Enabled)

    if(interface == NONE) {

        //if Not loggedIn -> Redir to Error.html
        cgi_redirectto(state, REDIR_HOST "error");
        return 0;

    } else {

        //Perform action & refresh index
        menu(newState);
        cgi_redirectto(state, REDIR_HOST);
        return 0;

    }

} //End CGI Common

int cgiBtRight(HttpState* state){
    return cgiBtCommon(state, buttonRight ); }

int cgiBtUp (HttpState* state){
    return cgiBtCommon(state, buttonUp ); }

int cgiBtDown (HttpState* state){
    return cgiBtCommon(state, buttonDown ); }

int cgiBtLeft (HttpState* state){
    return cgiBtCommon(state, buttonLeft ); }
```

Obrázek 3.11 - CGI Funkce - Stisk tlačítka

3.4 Ovládací Menu jednotky

Jádrem celého programu řídící jednotky je funkce *menu* potažmo Menu „Handler“. Ve skutečnosti se však nejedná o samostatný proces, ale o funkci která je shodně volána jak *ButtonsHandlerem* tak *HTTPHandlerem* (prostřednictvím CGI funkce *cgiBtCommon*)*. Tato funkce (a jí volané podfunkce) jednak obsahuje veškerý kód „vykreslování“ jednotlivých obrazovek menu, ovládání kursoru apod. a současně je v ní soustředěn veškerý výkonný kód, který skutečně mění jednotlivá nastavení řídící jednotky, ukládá, maže a mění programy zavlažování, atd. V této funkci je tedy realizováno vlastní plánování událostí uživatelem (zadávání úkolů pro řídící jednotku), které jsou touto funkcí uloženy a následně ve správný okamžik vykonány *TaskHandlerem*.

V této kapitole je dále popsáno provedení ovládacího menu jednotky, v první části z hlediska vnitřní funkce menu - způsob realizace Menu „Handleru“, v druhé části z hlediska vnější podoby prezentované uživateli.

3.4.1 Menu „Handler“

Funkce *menu* je vlastně jedním velkým stavovým automatem s každé její volání je přechodem do nového stavu. V závislosti na novém aktuálním stavu, jsou pak vykonány určité činnosti dané tímto stavem.

```

//***** Menu Handler *****/

void menu(uint nextState){
    //handles menu state
    //performs actual actions

    menuState = nextState; //Change State

    switch(menuState) {

//===== Menu States' Codes =====//

        case MS_VIEW_PROGRAM: ...

        ...

//=====//

    } //End Menu State Case
} //End Menu Handler
```

Obrázek 3.12 - Menu „Handler“ - Princip funkce

* V současné verzi programu je volána pouze *HTTPHandlerem* a *ButtonsHandler* volá funkci *buttonsOverride*, která přímo nastavuje manuální ovládání a to obdobným způsobem jako samotný Menu „Handler“.

Jednotlivé stavy vykonávají zpravidla některé z těchto činností: 1. zapsání nového obsahu na display - statický text nebo i proměnná data, 2. nastavení kurzoru, 3. načtení souvisejících nastavení do dočasných proměnných (nové hodnoty nastavení jsou uloženy vždy až po konečném potvrzení uživatelem), 4. změna nastavení v dočasných proměnných, 5. uložení nových nastavení či provedení akce, 6. nastavení sousedních stavů (do kterých se přechází při stisku jednoho ze 4 ovládacích tlačítek), 7. přímý přechod do jiného stavu.

Obrázek 3.13 ukazuje příklad kódu jednoduché obrazovky (Program View Screen). V tomto případě pro obsluhu obrazovky postačuje jediný stav menu. Při přechodu do tohoto stavu je na displej nejprve vypsán statický obsah, čímž se přepíše obsah displeje vypsany předcházející obrazovkou. Poté se vypisuje dynamický obsah (proměnná data), zde aktuální datum a čas na řádku displeje 0 a celé řádky 1 a 2, o které se stará funkce *refreshProgramScr*. Dále je provedeno nastavení kurzoru (v tomto případě skrytí) a nastavení sousedních stavů - obrazovek, do kterých přejde při stisku příslušného tlačítka.

```
case MS_VIEW_PROGRAM:
    //Display Write
    dlWrite(0,"xx 00.00.00 00:00:00");
    //Write(1,"Prog: xxxxxxxxxxxxxxxx");
    //Write(2,"Mode: yyyyyy      ");
    dlWrite(3,"Sett.  (1/4)   Menu");
    //Data Write
    //parse & write actTime
    mktm(&tmpTm,SEC_TIMER);
    writeDateTimeSecs(displLine[0],&tmpTm);
    //Actual program + mode
    refreshProgramScr();
    //Cursor
    setCursorOff();
    //SetNextStates
    buttonLeft  = MS_SETTINGS;
    buttonRight = MS_PROGRAM_MENU;
    buttonDown  = MS_VIEW_NEXT_STEP;
    buttonUp    = MS_VIEW_OVERRIDE;
    break;
```

Obrázek 3.13 - Menu „Handler“ - Ukázka jednostavové obrazovky

Obrázek 3.14 ukazuje konstrukci o něco složitější obrazovky (menu Measure Settings) skládající se již ze 4 stavů. Při přechodu na tuto obrazovku, je v inicializačním stavu opět přepsán obsah displeje, nastaven režim kurzoru a poté inicializována dočasná proměnná *tmpInt* (ta je zde používána jako index aktuální podobrazovky) a nastaveny sousední stavy, kde MS_MEASUREU a MS_MEASURED jsou pomocné stavy této obrazovky. Následně se vykoná ještě kód stavu MS_MEASURE1 (refresh zobrazení), který vypíše na displej dynamický obsah - index a obsah podobrazovky (zde jméno následujícího menu). Refresh stav ještě nastaví pravý sousední stav obrazovky.

Po stisknutí tlačítka „nahoru“ nebo „dolu“ je změna hodnota proměnné *tmpInt* (změna podobrazovky) a okamžitě se přejde zpět do refresh stavu, který aktualizuje obsah displeje a nastaví nový stav pro pravé tlačítko.

```
//=> ----- Measure Settings Menu ----- <=//
case MS_MEASURE: //(init state)
//Display Write
dlWrite(0,"Measure Settings:  ");
//Write(1,"
//Write(2,"
dlWrite(3,"<<      (x/3)      >>");
//Cursor init
setCursorOff();
tmpInt = 1; //init subscreen counter
//set back state & slave states
buttonLeft = MS_SETTINGS;
//buttonRight =
buttonDown = MS_MEASURED;
buttonUp   = MS_MEASUREU;

case MS_MEASURE1: //(refresh state)
//Refresh scr counter
writeNum( (displine[3]+8),tmpInt,1);
//Refresh subscreen
switch(tmpInt) {
    case 1: dlWrite(1,"  Global Settings  ");
            buttonRight = MS_MEASURE_SET; break;
    case 2: dlWrite(1,"  Sensors' Settings ");
            buttonRight = MS_SENSORS;    break;
    case 3: dlWrite(1,"  Clear History   ");
            buttonRight = MS_HISTORY_CLEAR;
    } break;

//Slave States
case MS_MEASURED: //(Button Dwon)
//increase screen counter
if(tmpInt == 3) tmpInt = 1; //Rollover
else tmpInt++;
//Invoke refresh
menu(MS_MEASURE1);
break;

case MS_MEASUREU: //(Button Up)
//decrease screen counter
if(tmpInt == 1) tmpInt = 3; //Roll"under"
else tmpInt--;
//Invoke refresh
menu(MS_MEASURE1);
break;
```

Obrázek 3.14 - Menu „Handler“ - Ukázka složitější obrazovky

Nejsložitější obrazovky programu (zpravidla Settings Screeny, které pracují s několika parametry nastavení najednou) sestávají až z desítek stavů Menu „Handleru“.

3.4.2 Koncepce ovládacího menu

Ovládací menu řídicí jednotky je realizováno prostřednictvím 4 směrových tlačítek a displeje 4x20 znaků. Z toho vyplývá pojetí menu, kdy se uživatel pomocí tlačítek naviguje mezi množstvím různých obrazovek. Jednotlivé obrazovky pak slouží buďto k prezentaci dat a stavu jednotky, nebo ke změně nastavení jednotky.

„Mapa“ uspořádání ovládacího menu a konkrétní popis jeho jednotlivých obrazovek je umístěn v příloze **Struktura menu**. V této kapitole je blíže popsána forma zmíněné přílohy a uveden obecný popis koncepce menu a jeho obrazovek (typy View Screen, Menu a Settings Screen).

Obrazovky obecně: Aby byla uživateli usnadněna orientace v menu, je ve většině obrazovek ze čtyř řádků displeje „obětován“ alespoň první řádek - pro název aktuálního menu/obrazovky a řádek poslední - pro naznačení funkce jednotlivých tlačítek.

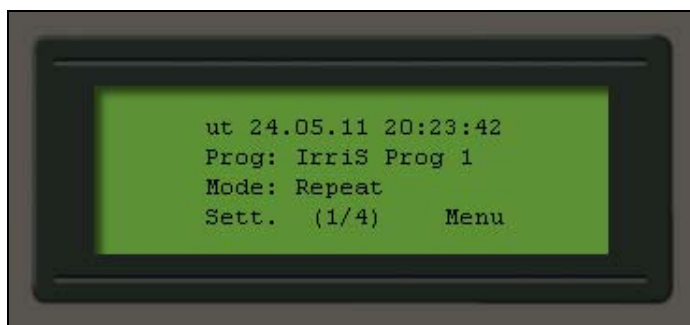
Tlačítka „nahoru“ a „dolu“ se nejčastěji používají buďto pro změnu nějaké hodnoty či nastavení (indikováno zobrazením kurzoru), nebo pro vertikální navigaci mezi položkami menu, potom je uprostřed 4. řádku displeje zobrazeno například „(1/3)“, kde 1 znamená pořadí aktuální položky a 3 celkový počet položek. Vertikální navigací na obrazovkách typ Menu je možné přecházet mezi obrazovkami, které jsou v příloze **Struktura menu** uvedeny na stejné úrovni (stejně odsazení) v rámci daného menu.

Tlačítka „vpravo“ a „vlevo“ se nejčastěji používají pro posun kurzoru (pokud je zobrazen), popřípadě i pro horizontální navigaci mezi jednotlivými obrazovkami u víceobrazovkových nastavení. Tyto možnosti jsou indikovány zobrazením symbolů „<<“ nebo „>>“ na krajích 4. řádku displeje (platí pro levé a pravé tlačítko zvlášť). Obdobně platí i pro navigaci v menu - pravé tlačítko se využívá pro vstup („zanoření“ hlouběji) do menu (v příloze **Struktura menu** značeno symbolem „→“) a levé tlačítko pro návrat o úroveň výš.

Pravé tlačítko je také používáno pro potvrzení změny nastavení (indikováno zobrazením „Confirm“ v pravém dolním rohu displeje) a levé tlačítko pro zrušení změny nastavení (zobrazení „Cancel“ v levém dolním rohu) nebo pro ukončení menu a návrat do výchozího View Screenu (zobrazení „Exit“).

Další funkce lokálního ovládacího rozhraní: (Tyto funkce nejsou ve webovém ovládacím rozhraní z principu možné.) Funkce stisku a podržení tlačítka není v tuto chvíli podporována, stisk více tlačítek najednou deaktivuje manuální ovládání.*

Obrazovky typu View Screen: (v příloze [Struktura menu](#) jsou značeny stylem „*Program View*“) Tyto obrazovky slouží výhradně k zobrazování údajů o aktuálním stavu zavlažovacího systému a jsou základem celého menu (nejvyšší úroveň). Obrazovky jsou celkem 4 a lze se mezi nimi posouvat pomocí tlačítek „nahoru“ a „dolu“, tlačítko vlevo přechází do nabídky Menu Settings, tlačítko vpravo do ostatních nabídek podle konkrétního View Screenu (podrobněji viz [Struktura menu](#)).



Obrázek 3.15 - Ukázka obrazovky typu View Screen

Obrazovky typu Menu: (v příloze [Struktura menu](#) značeny stylem „*Settings Menu*“) Slouží čistě pro navigaci mezi ostatními obrazovkami. Výběr například zavlažovacího programu, kroku programu či prohlížení historie měření se také chová jako obrazovka tohoto typu. V těchto obrazovkách jsou tlačítka „nahoru“ a „dolu“ použita pro posouvání mezi položkami menu, tlačítko „vpravo“ pro „zanoření“ do dalšího menu a tlačítko vlevo pro návrat o úroveň výš.



Obrázek 3.16 - Ukázka obrazovky typu Menu

* V původní koncepci ovládání měla být funkce držení tlačítka povolena u tlačítek „nahoru“ a „dolu“ pro zrychlení změn hodnot (s využitím časovače A) a stisk jakýchkoli dvou a více tlačítek ignorován.

Obrazovky typu Settins Screen: (v příloze **Struktura menu** značeny stylem „**Change Program**“) V těchto obrazovkách dochází k samotným změnám nastavení řídicí jednotky zavlažování. Jsou charakteristické zobrazením kurzoru (ve webovém rozhraní podtržený znak). Tlačítka „nahoru“ a „dolu“ jsou použita pro změnu nastavované hodnoty nebo parametru (na kterém je právě kurzor), popřípadě pro posun mezi nastavovanými parametry (pokud je kurzor na hlavičce parametru), někdy i mezi více obrazovkami u rozsáhlejších nastavení (vertikální navigace). Tlačítka „vlevo“ a „vpravo“ jsou použita pro posuv kurzoru, někdy i mezi více obrazovkami u rozsáhlejších nastavení (horizontální navigace). Pokud je kurzor na levém kraji obrazovky tlačítko „vlevo“ se používá pro zrušení změn nastavení. Pokud je kurzor na pravém kraji obrazovky tlačítko vpravo se používá pro uložení nového nastavení.



Obrázek 3.17 - Ukázka obrazovky typu Settings Screen

Bližší informace ke konkrétním obrazovkám nastavení a jejich uspořádání v menu viz příložená **Struktura menu**.

3.5 Shrnutí

Stávající program umožňuje tyto funkce řídicí jednotky:

- zobrazení aktuálního data a času,
- zobrazení relativních vlhkostí ovzduší aktuálně měřených až 4 čidly,
- ukládání a možnost prohlížení historie relativních vlhkostí naměřených všemi 4 čidly (až 99 položek), možnost ukládání průměrných hodnot z několika vzorků (počet nastavitelný),
- zobrazení nastaveného zavlažovacího programu, jeho režimu a následující změny stavu zavlažovacího systému i seznamu jednotlivých zavlažovacích kroků s jejich plánovanými daty a časy,
- možnost uložení až 9 zavlažovacích programů, každý po 50 zavlažovacích krocích ovládajících všechny 4 ventily v libovolných kombinacích,
- libovolné nastavování data a času plánovaných akcí (po 1 s), intervaly nastavitelné libovolně od 30 s do 30 dnů (po 1 s),
- režim cyklického opakování programu,
- možnost manuálního ovládání všech 4 ventilů (závlaha navíc, přestávka v závlaze) s platností 30 s až 30 dní (po 1 s),
- nastavitelná funkce podsvícení displeje a množství dalších praktických nastavení (přednastavené délky intervalů, řízení spotřeby, seřízení citlivosti čidel, apod.),
- Zachování správného reálného času a uložených dat a nastavené i při dlouhodobém výpadku napájení.

Program jednotky by v budoucnu mohl být doplněn o tyto funkce a vylepšení:

- statistika relativní vlhkosti - výpočet klouzavého průměru vlhkosti ovzduší v dlouhodobějším časovém měřítku,
- dílčí zavlažovací programy, nastavitelné samostatně pro jednotlivé zavlažovací okruhy,
- možnost zřetězení zavlažovacích programů,
- možnost větvení zavlažovacích programů v závislosti na statistice vlhkosti ovzduší,
- nastavitelná sezónní závlaha (v procentech oproti původní zálivce),
- komplexnější manuální ovládání - možnost dílčího ovládání jednotlivých okruhů a možnost naplánování více kroků do budoucnosti,
- možnost zálohování a obnovy nastavení a dat řídicí jednotky přes webové ovládací rozhraní,
- ověření identity uživatele (username, password) popřípadě i správa oprávnění uživatelů,
- vylepšení způsobu ukládání a správy paměti, statistika využití paměti,
- optimalizace webového rozhraní i pro další webové prohlížeče a jiná rozlišení plochy,
- možnost dynamického nastavení IP adresy zařízení.

ZÁVĚR

Při návrhu řídicí jednotky bylo dbáno zejména na možnost jejího univerzálního použití. Obvodová řešení byla volena co nejjednodušší při zachování všech požadavků zadání a důrazu na soustředění řízení do centrální procesoru. Při návrhu konstrukce bylo dbáno na celkovou elegantnost a kompaktnost finálního zařízení.

Výsledkem práce je malá řídicí jednotka určená pro zavlažovací systém, která umožňuje nezávislé spínání čtyř libovolných elektroventilů až do napájení 250V/10A AC a připojení čtyř nezávislých snímačů vlhkosti s napěťovým výstupem, pro které je poskytnuto napájení 5 V. Obvod reálného času a uložená data jsou zálohována malou plochou baterií. Na předním panelu zařízení je kontrolka zapnutí, kontrolky stavu ventilů a lokální ovládací rozhraní, realizované čtyřmi směrovými tlačítky a LCD displejem 4x20 znaků. Kvůli internímu bugu použitého procesoru se však nepovedlo displej oživit, proto jsou v současné verzi zařízení tlačítka využita pouze k manuálnímu ovládání ventilů a podsvícení displeje k nastavitelné indikaci událostí. Přesto zařízení pracuje správně a všechny funkce jsou v plném rozsahu dostupné přes Ethernetovou přípojku na webovém ovládacím rozhraní.

Jádrem programu řídicí jednotky je funkce Menu zapouzdřující veškerý kód ovládacích obrazovek a funkcí programu. Tato funkce je uzpůsobena pro snadné přidávání nových obrazovek a funkcí a je určena pro využití jak lokálním tak vzdáleným ovládáním, změny v ní se tedy projeví shodně na obou ovládacích rozhraních. Současný program jednotky umožňuje měření a ukládání historie vlhkosti ovzduší, správu uživatelských programů zavlažování a řízení připojených ventilů těmito programy nebo manuálním ovládáním.

Vyrobené zařízení splňuje požadavky původního zadání. V budoucnu by však bylo dobré opravou zapojení LCD displeje zprovoznit lokální ovládání v plném rozsahu a případně doplnit program jednotky o další pokročilejší funkce.

LITARATURA A ZDROJE INFORMACÍ

- [1] AZ Zahrada. *Zavlažovací systémy a závlahy* [online]. [cit. 14. prosinec 2010]. Dostupné na: <<http://azzahrada.cz/>>.
- [2] AZ Zahrada. *Komponenty závlahových systémů* [online]. [cit. 14. prosinec 2010]. Dostupné na: <<http://azzahrada.cz/zavlazovaci-systemy/komponenty-zavlahovych-systemu.php>>.
- [3] TME Czech Republic s.r.o.. *Online katalog Transfer Multisort Elektronik: snímače vlhkosti* [online]. [cit. 8. listopad 2010]. Dostupné na: <http://www.tme.eu/cz/katalog/index.phtml#id_category%3D100525%26cleanParameters%3D1>.
- [4] GM Electronic, spol. s r. o.. *Katalog: Součástky pro elektroniku 2005*.
- [5] GM Electronic, spol. s r. o.. *E-shop: Elektronické součástky* [online]. [cit. listopad 2010]. Dostupné na: <<http://www.gme.cz/cz/elektronicke-soucastky/>>.
- [6] DatasheetCatalog.com. *Datasheety k součástkám* [online]. [cit. listopad 2010]. Dostupné na: <<http://datasheetcatalog.com/>>.
- [7] Rabbit Semiconductor. *Popis a manuály k modulu Rabbit RCM3200* [online]. [cit. 3. listopad 2010]. Dostupné na: <<http://www.rabbit.com/products/rcm3200/docs.shtml>>.
- [8] Rabbit Semiconductor. *Dynamic C[®] TCP/IP User's Manual* [online]. [cit. 3. listopad 2010]. Dostupné na: <<http://www.rabbit.com/>>.
- [9] Vorel, P.; Patočka, M. *Průmyslová elektronika*. Fakulta elektrotechniky a komunikačních technologií, VUT Brno, srpen 2007.
- [10] Rabbit Semiconductor. *TN228: Paralel Port F Bug* [online]. [cit. 27. duben 2011]. Dostupné na: <ftp1.digi.com/support/documentation/0220076_a.pdf>.

SEZNAM POUŽITÝCH ZKRATEK

AC	Alternating Current
CGI	Common Gateway Interface
DC	Direct Current
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Procotol
PWM	Pulse Width Modulation
URL	Uniform Resource Locator
SSI	Server Side Includes
TCP	Transmission Control Protocol
AZS	Automatický závlahový systém
DPS	Deska plošných spojů

SEZNAM OBRÁZKŮ

1 AUTOMATICKÉ ZAVLAŽOVACÍ SYSTÉMY

Obrázek 1.1 - Ukázka rotačního postřikovače - Hunter PGM.....	11
Obrázek 1.2 - Funkce elektroventilu	12
Obrázek 1.3 - Správná instalace elektroventilů.....	13
Obrázek 1.4 - Ukázka řídicí jednotky - Hunter EC	14

2 NÁVRH FYZICKÉ REALIZACE JEDNOTKY

Obrázek 2.1 - Řídicí modul Rabbit RCM3200.....	18
Obrázek 2.2- Zobrazovací modul CM200400SFAYA-02	20
Obrázek 2.3 - Základní připojení řídicího modulu	21
Obrázek 2.4 - Vstupní obvod	23
Obrázek 2.5 - Výstupní obvod (jeden výstup).....	23
Obrázek 2.6 - Ovládací tlačítka	24
Obrázek 2.7 - Zapojení zobrazovacího modulu	24
Obrázek 2.8 - Napájecí zdroj.....	25

3 NÁVRH ŘÍDICÍHO PROGRAMU JEDNOTKY

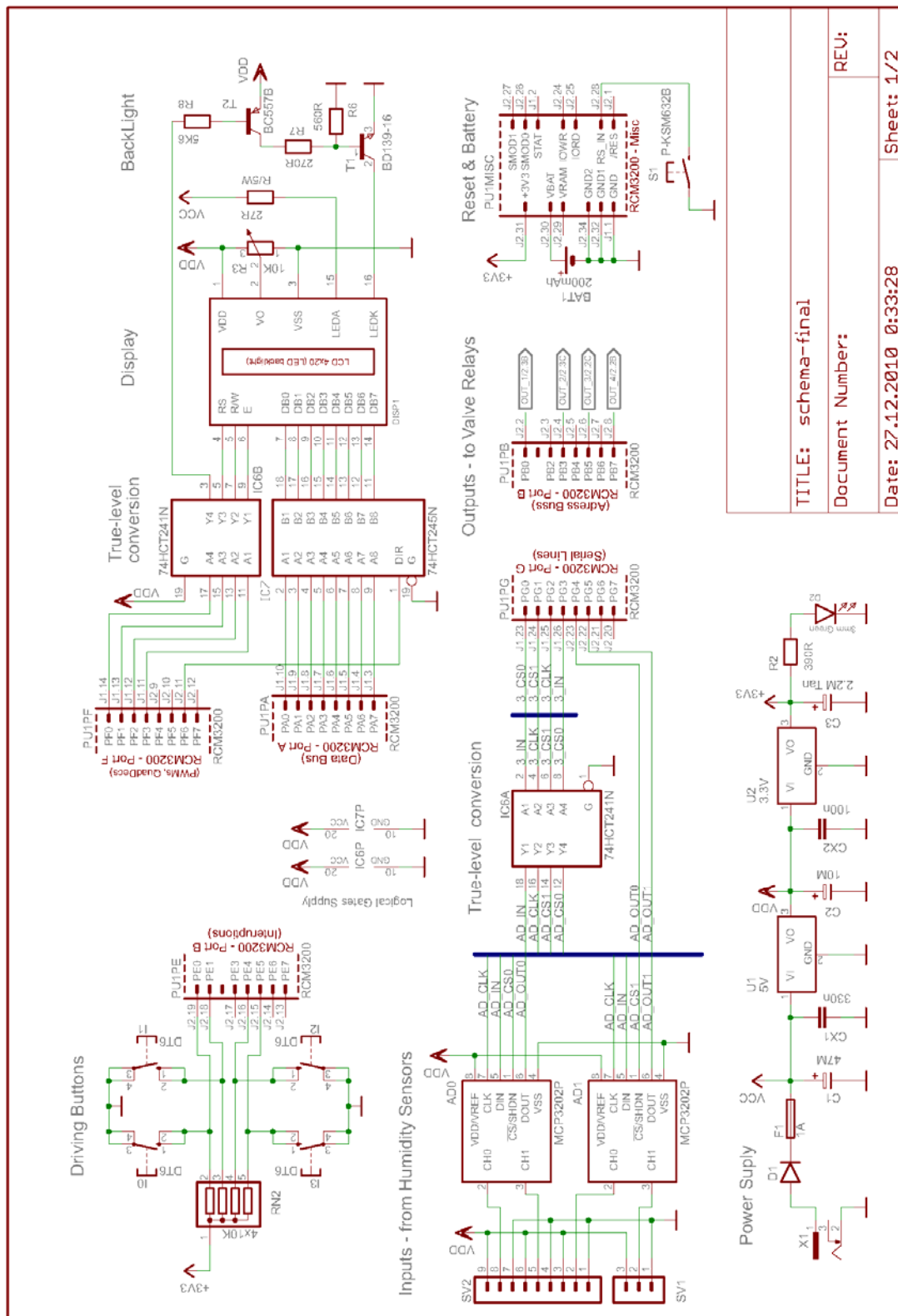
Obrázek 3.1 - Hlavní kód programu.....	33
Obrázek 3.2 - Buttons Handler.....	36
Obrázek 3.3 - Inicializace HTTP Serveru	37
Obrázek 3.4 - Task Handler - Správa timeoutů.....	39
Obrázek 3.5 - Task Handler - Refresh zobrazení	40
Obrázek 3.6 - Task Handler - Měření vlhkostí.....	41
Obrázek 3.7 - Task Handler - Aktualizace výstupů.....	42
Obrázek 3.8 - CGI Funkce - Přihlášení	45
Obrázek 3.9 - CGI Funkce - Odhlášení	45
Obrázek 3.10 - CGI Funkce - Generování hlavní stránky	46
Obrázek 3.11 - CGI Funkce - Stisk tlačítka	47
Obrázek 3.12 - Menu „Handler“ - Princip funkce.....	48
Obrázek 3.13 - Menu „Handler“ - Ukázka jednostavové obrazovky	49
Obrázek 3.14 - Menu „Handler“ - Ukázka složitější obrazovky	50
Obrázek 3.15 - Ukázka obrazovky typu View Screen.....	52
Obrázek 3.16 - Ukázka obrazovky typu Menu.....	52
Obrázek 3.17 - Ukázka obrazovky typu Settings Screen	53

SEZNAM PŘÍLOH

PŘÍLOHA A: TECHNICKÁ DOKUMENTACE	61
<i>A.1 Obvodové schéma - hlavní část.....</i>	<i>61</i>
<i>A.2 Obvodové schéma - část výstupů.....</i>	<i>62</i>
<i>A.3 Výkresy plošných spojů</i>	<i>63</i>
<i>A.4 Osazovací výkresy</i>	<i>64</i>
<i>A.5 Výkres propojení desek</i>	<i>65</i>
PŘÍLOHA B: FOTODOKUMENTACE.....	66
<i>B.1 Desky plošných spojů</i>	<i>66</i>
<i>B.2 Součástky.....</i>	<i>66</i>
<i>B.3 Osazené desky</i>	<i>66</i>
<i>B.4 Sestavený blok elektroniky</i>	<i>67</i>
<i>B.5 Díly konstrukční krabičky</i>	<i>67</i>
<i>B.6 Sestavená krabička.....</i>	<i>67</i>
<i>B.7 Zařízení v provozu.....</i>	<i>68</i>
PŘÍLOHA C: WEBOVÉ OVLÁDACÍ ROZHRANÍ.....	69
<i>C.1 Přihlašovací stránka</i>	<i>69</i>
<i>C.2 Hlavní ovládací stránka</i>	<i>69</i>
<i>C.3 Stránka po odhlášení.....</i>	<i>70</i>
<i>C.4 Chyba přihlášení.....</i>	<i>70</i>
PŘÍLOHA D: STRUKTURA MENU	71
PŘÍLOHA E: OBSAH PŘÍLOŽENÉHO CD	73

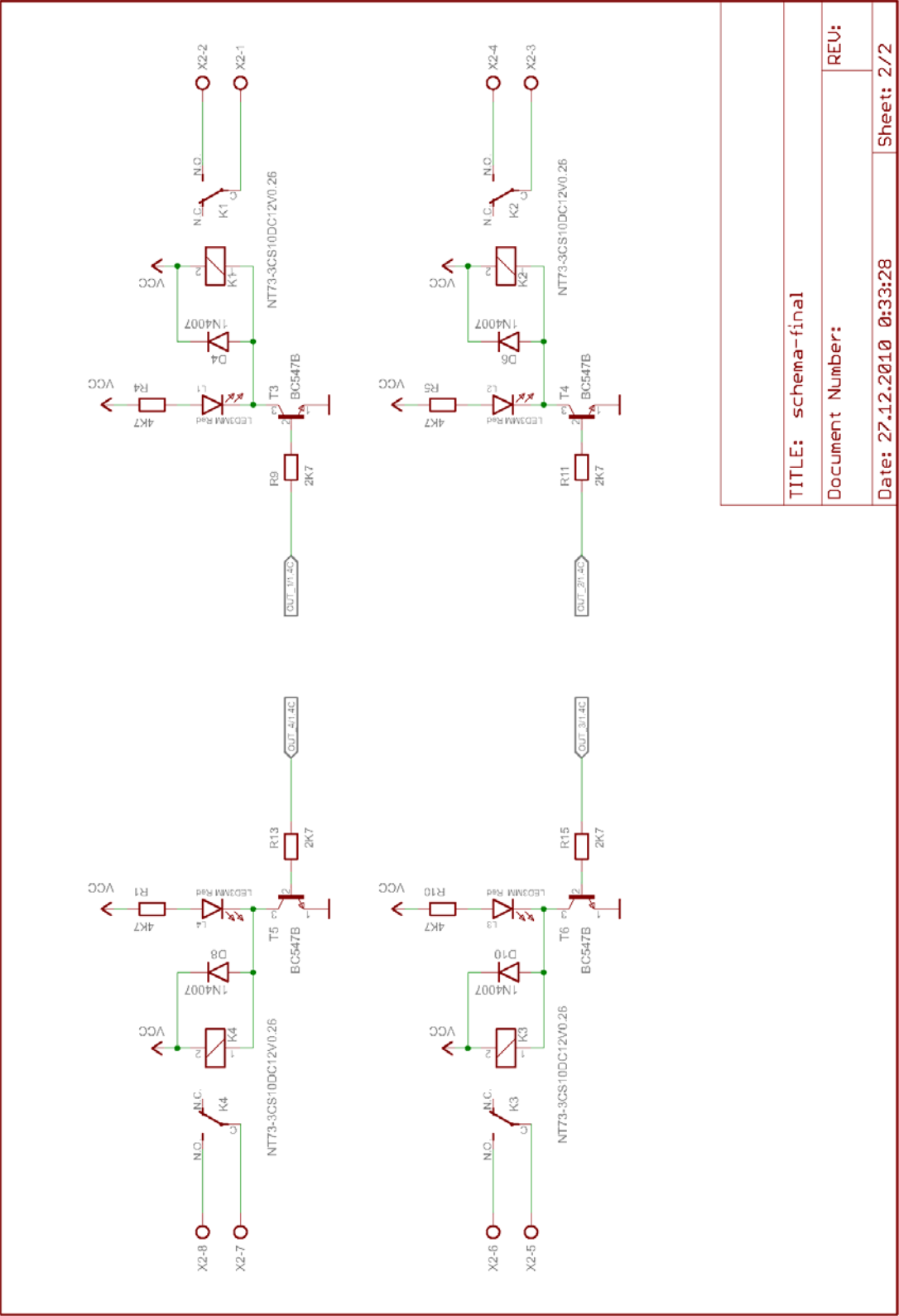
PŘÍLOHA A: TECHNICKÁ DOKUMENTACE

A.1 Obvodové schéma - hlavní část



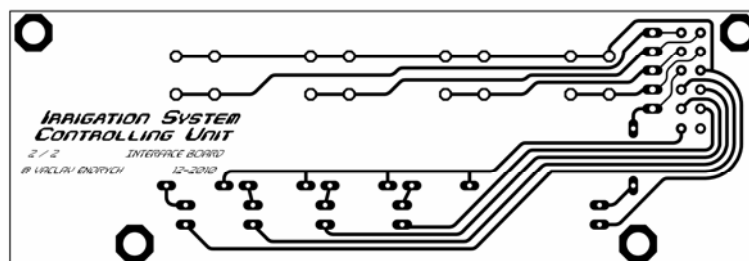
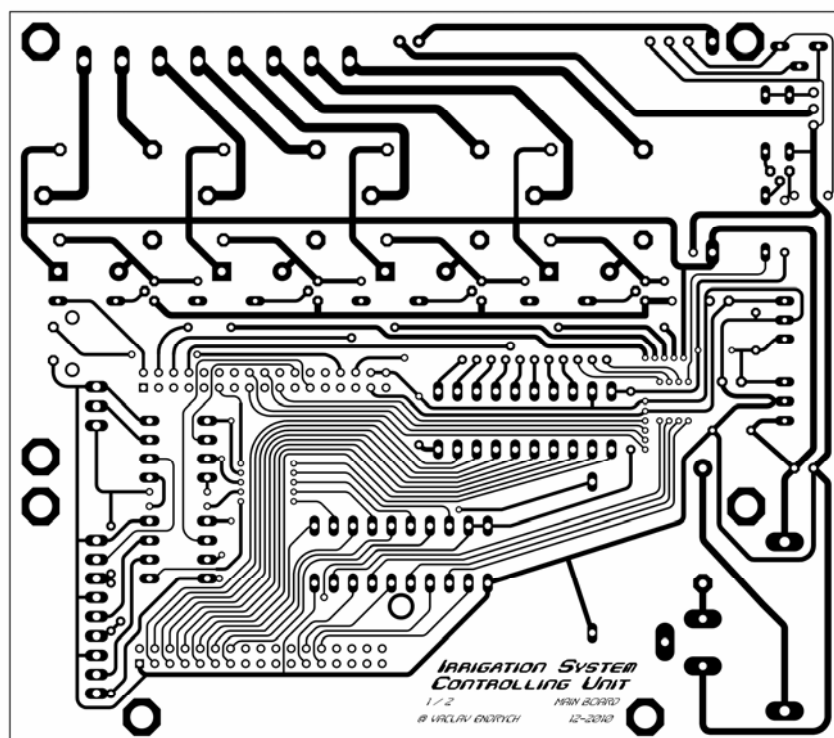
PŘÍLOHA A: TECHNICKÁ DOKUMENTACE

A.2 Obvodové schéma - část výstupů



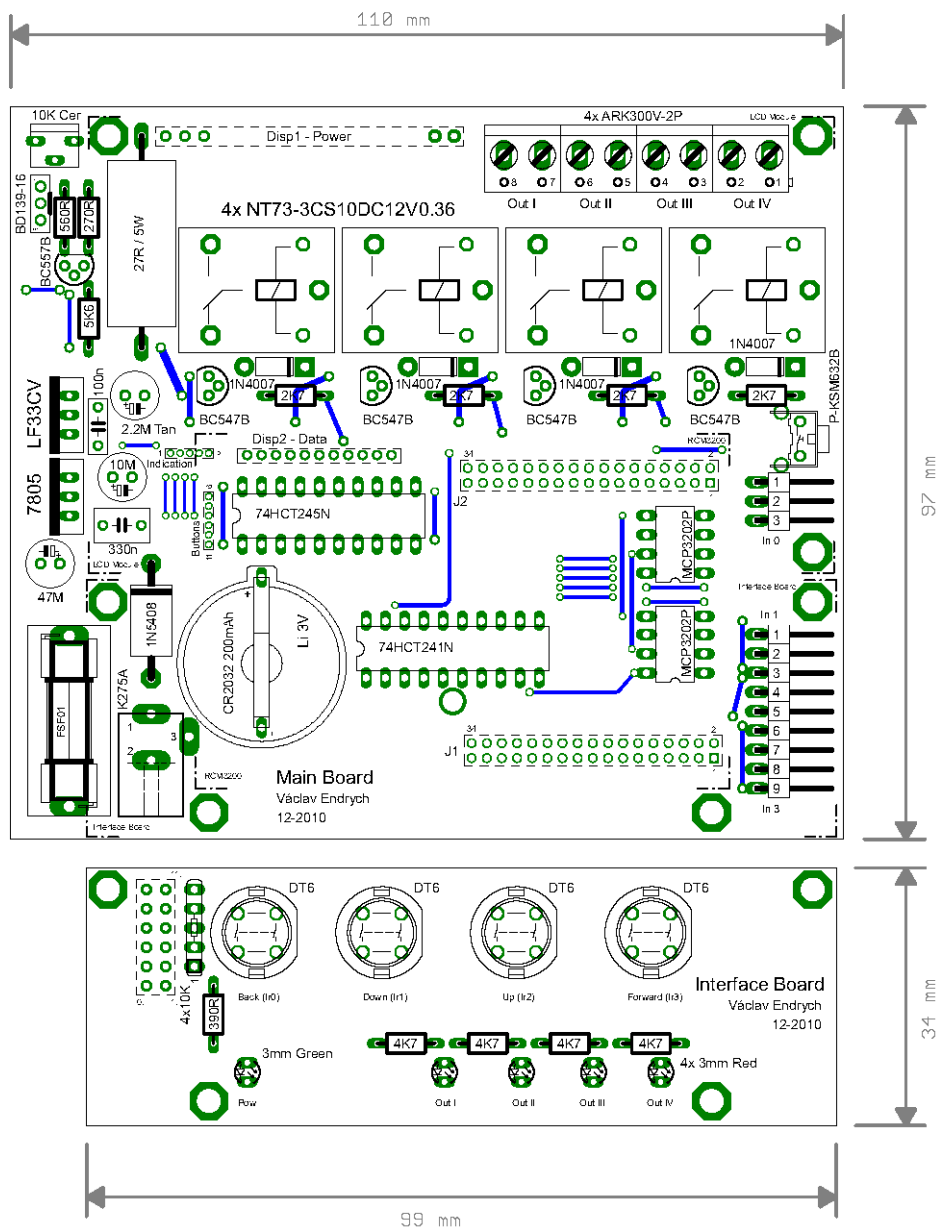
PŘÍLOHA A: TECHNICKÁ DOKUMENTACE

A.3 Výkresy plošných spojů



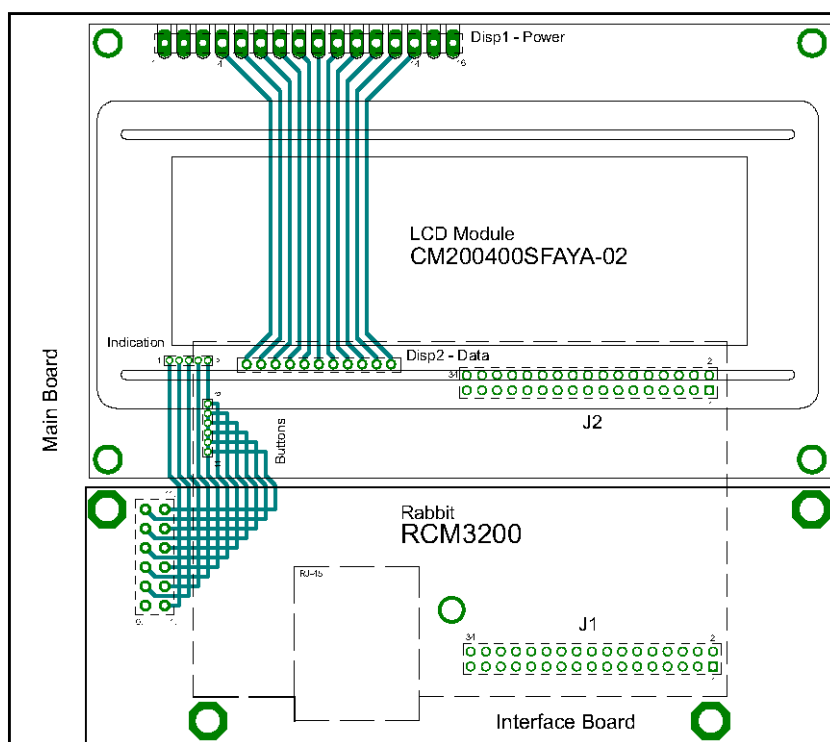
PŘÍLOHA A: TECHNICKÁ DOKUMENTACE

A.4 Osazovací výkresy



PŘÍLOHA A: TECHNICKÁ DOKUMENTACE

A.5 Výkres propojení desek



PŘÍLOHA B: FOTODOKUMENTACE

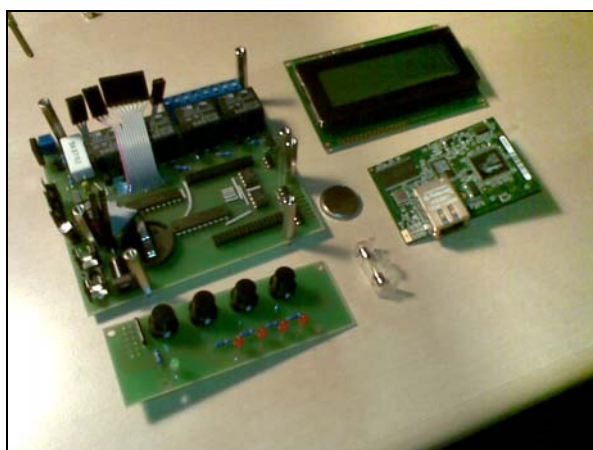
B.1 Desky plošných spojů



B.2 Součástky



B.3 Osazené desky



PŘÍLOHA B: FOTODOKUMENTACE

B.4 Sestavený blok elektroniky



B.5 Díly konstrukční krabičky



B.6 Sestavená krabička



PŘÍLOHA B: FOTODOKUMENTACE

B.7 Zařízení v provozu



PŘÍLOHA C: WEBOVÉ OVLÁDACÍ ROZHRANÍ

C.1 Přihlašovací stránka



C.2 Hlavní ovládací stránka

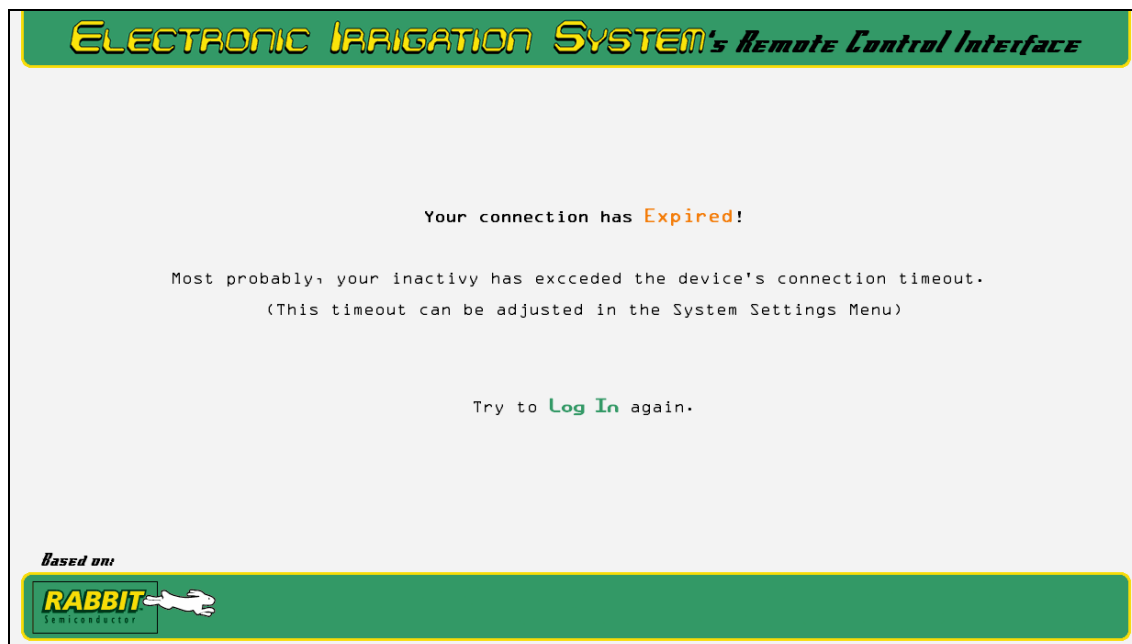


PŘÍLOHA C: WEBOVÉ OVLÁDACÍ ROZHRANÍ

C.3 Stránka po odhlášení



C.4 Chyba přihlášení



PŘÍLOHA D: STRUKTURA MENU

(Bližší popis této přílohy je uveden v kapitole **Koncepce ovládacího menu**)

Program View - Zobrazuje název aktivního programu zavlažování (nebo „(none)“, pokud není aktivní žádný program) a nastavený režim provedení programu (jednorázově nebo opakovaně).

→ *Settings Menu* (popsáno na konci přílohy)

→ *Program Menu*

→ **Program change** - Na této obrazovce lze změnit aktivní program, nebo také deaktivovat právě nastavený program, dále změnit režim provedení programu nebo nastavit nový následující krok programu včetně data jeho provedení.

→ *View/Edit program*

→ *View/Edit Header Data* - V tomto menu lze prohlížet a editovat jednotlivá hlavičková data programu: jméno, datum počátku a délku. Nastavení délky programu je omezeno podle obsažených kroků.

→ *View/Edit Program Steps* - V tomto menu lze prohlížet a editovat jednotlivé kroky programu. Dále lze kroky vymazat nebo přidat nové.

→ **Clear all Steps** - Zde lze vymazat všechny kroky programu najednou.

→ **Delete Program** - Tato volba odstraní celý zavlažovací program.

→ **Add New Program** - Zde lze vytvořit nový (prázdný) zavlažovací program, je pak třeba nastavit jméno programu, počáteční datum a délku programu.

Next Step View - Zobrazuje nejbližší následující krok aktivního programu včetně jeho plánovaného data, nebo „(none)“ pokud není aktivní žádný program, nebo „(Repeat)“ či „(ProgEnd)“ po provedení posledního kroku aktivního programu.

→ *Settings Menu* (popsáno na konci přílohy)

→ *View/Edit Program Steps* (Pokud je aktivní nějaký program, je zde možné přejít přímo k jeho jednotlivým krokům, podobně jako v nabídce popsané výše.)

Measure View - Zobrazuje hodnoty vlhkosti ovzduší aktuálně měřené připojenými čidly. Hodnoty jsou obnoveny každou 1 s, převody na relativní vlhkost jsou provedeny podle nastavení jednotlivých čidel v menu Measure Settings. Pro čidla nastavená jako „Not Connected“ je zobrazována hodnota „(N/C)“.

→ *Settings Menu* (popsáno na konci přílohy)

→ *Measure History* - Zde lze prohlížet historii vlhkosti ovzduší naměřenou dle nastavení v menu Measure Settings.

PŘÍLOHA D: STRUKTURA MENU *(Pokračování)*

Manual Override View - Zobrazuje stav manuálního (nadřazeného) ovládání - „Inactive“, pokud je vypnuté, stavy výstupů jednotky jsou pak ovládány zavlažovacím programem (pokud je nějaký aktivní), nebo „Active until“ včetně data vypršení platnosti, pokud je ovládání zapnuté, stav výstupů jednotky je pak pevně dán manuálním ovládáním (je vidět na kontrolkách výstupů).

→ *Settings Menu* (popsáno na konci přílohy)

→ **Activate Manual Override** - Pokud je manuální ovládání neaktivní, je zde možné ho aktivovat. Pak je třeba nastavit požadovaný stav výstupů a dobu platnosti.

→ **Deactivate Manual Override** - Pokud je manuální ovládání aktivní je zde možné jej deaktivovat.

Settings Menu

→ **Menu Settings** - Zde je možné nastavit výchozí obrazovku menu (Program View, Next Step View nebo Measure View) a příznak, zda se má jako výchozí zobrazit Manual Override View v případě, že je aktivní manuální ovládání.

→ **Program Settings** - Zde lze změnit přednastavenou délku trvání zavlažovacích programů a manuálního ovládání (implicitně použitou při aktivaci pomocí lokálního ovládacího rozhraní).

→ **Display Settings** - V této nabídce lze nastavit jas podsvícení LCD displeje, režim indikace (Buttons Activity, HTTP Activity, Connected) a timeout (délku rozsvícení) pro režimy „Activity“.

→ *Measure Settings*

→ **Global Measure Settings** - Zde se dá nastavit datum a čas nejbližšího následujícího měření vlhkosti a interval pro měření vlhkostí a také průměrování (počet vzorků, ze kterých se má vypočíst průměrná hodnota k uložení do historie).

→ *Sensors' Settings* - V tomto menu lze prohlížet a měnit nastavení jednotlivých čidel vlhkosti.

→ **Clear Measure History** - V této nabídce lze vymazat všechny záznamy historie měření vlhkosti ovzduší.

→ **System Settings** - Zde lze změnit taktovací frekvenci řídicího procesoru jednotky a timeout přihlášení k webovému ovládacímu rozhraní.

→ **Set Time** - V této obrazovce se nastavuje aktuální datum a čas.

PŘÍLOHA E: OBSAH PŘILOŽENÉHO CD

<root>/ElectronicIrrigationSystem.pdf

elektronická verze této bakalářské práce

<root>/SourceCode.c

zdrojový kód programu řídicí jednotky

<root>/Pages/

ukázka stránek webového ovládacího rozhraní (popis viz kapitola 3.3)

<root>/Pages/pics/

obrázky pro webové rozhraní

<root>/Pages/fonts/

použité fonty